

27 InfoProvider

Der Zugriff auf Datenziele mit physisch vorhandenen Daten erfüllt nur einen Teil der Anforderungen, die an die Analytical Engine gestellt werden. Ebenso wichtig kann es sein, bei der Analyse von Daten mehrere physische Datenbestände in einen gemeinsamen Kontext zu stellen oder Daten sogar aus ganz anderen Quellen zu beziehen.

Zu diesem Zweck bietet das BW über die Datenziele hinaus weitere InfoProvider, die lediglich als Metadatenobjekte definiert werden und Lesezugriffe an die Datenstrukturen physischer Datenziele richten. Je nachdem, welches Ziel mit dem Einsatz eines solchen InfoProviders verfolgt wird und welche Datenquellen zur Analyse herangezogen werden sollen, können unterschiedliche Typen von InfoProvider verwendet werden:

- **MultiProvider**, um Daten mehrerer InfoProvider in einen gemeinsamen Kontext zu stellen (Abschnitt 27.1),
- **semantisch partitionierte Objekte**, um die Daten aus den Partitionen semantisch partitionierter Objekte in einen gemeinsamen Kontext zu stellen (Abschnitt 27.2),
- **InfoSets**, um Daten von InfoCubes, DataStore-Objekten und Info-Objekten mit Stammdaten relational zu verknüpfen (Abschnitt 27.3),
- **TransientProvider**, um ohne vorangehende Datenmodellierung Ad-hoc-Analysen (auch im SAP ERP) auszuführen oder um Daten in analytischen Indizes zu analysieren (Abschnitt 27.4),
- **CompositeProvider**, um die Inhalte unterschiedlicher analytischer Indizes zusammenzuführen und ggf. mit Daten »normaler« InfoProvider des BW zu verbinden (Abschnitt 27.5),
- **VirtualProvider** mit Funktionsbaustein und BAPI zur Implementierung spezifischer Analyseanforderungen, die nur durch eigendefinierte Programmlogik erfüllt werden können (Abschnitt 27.6).

Die Definition und Funktionsweise dieser InfoProvider wird in den nachfolgenden Abschnitten erläutert.

27.1 MultiProvider

MultiProvider dienen dazu, die Daten unterschiedlicher InfoProvider zusammenzuführen und damit in einen gemeinsamen Kontext zu stellen. Dies kann aus unterschiedlichen Gründen erforderlich sein:

- Es existieren mehrere InfoProvider, die betriebswirtschaftlich abgeschlossene Bereiche darstellen. Diese verfügen zwar über eine gemeinsame Schnittmenge mit identischem Zeitbezug (zum Beispiel enthalten alle InfoProvider eine Kundennummer und beziehen ihre Daten auf eine Geschäftsperiode), haben aber ansonsten eine heterogene Datenstruktur. Die Merkmale, die die Schnittmenge bilden, sollen in einen gemeinsamen Kontext gestellt werden, um sie im Reporting zusammenzuführen.
- Es sollen die Daten unterschiedlicher InfoProvider, die aus technischen Gründen nicht in einem InfoProvider vorgehalten werden (zum Beispiel weil strukturähnliche Daten auf InfoCubes, virtuelle Cubes und DataStore-Objekte verteilt sind), in einen gemeinsamen Kontext gestellt werden.
- Es sollen Daten, die aus Designgründen auf mehrere InfoProvider verteilt sind (zum Beispiel Plan- und Istdaten) in einen gemeinsamen Kontext gestellt werden.
- Daten wurden zur Verbesserung der Performance auf mehrere InfoProvider verteilt und müssen zur Analyse der Daten wieder zusammengefügt werden¹.



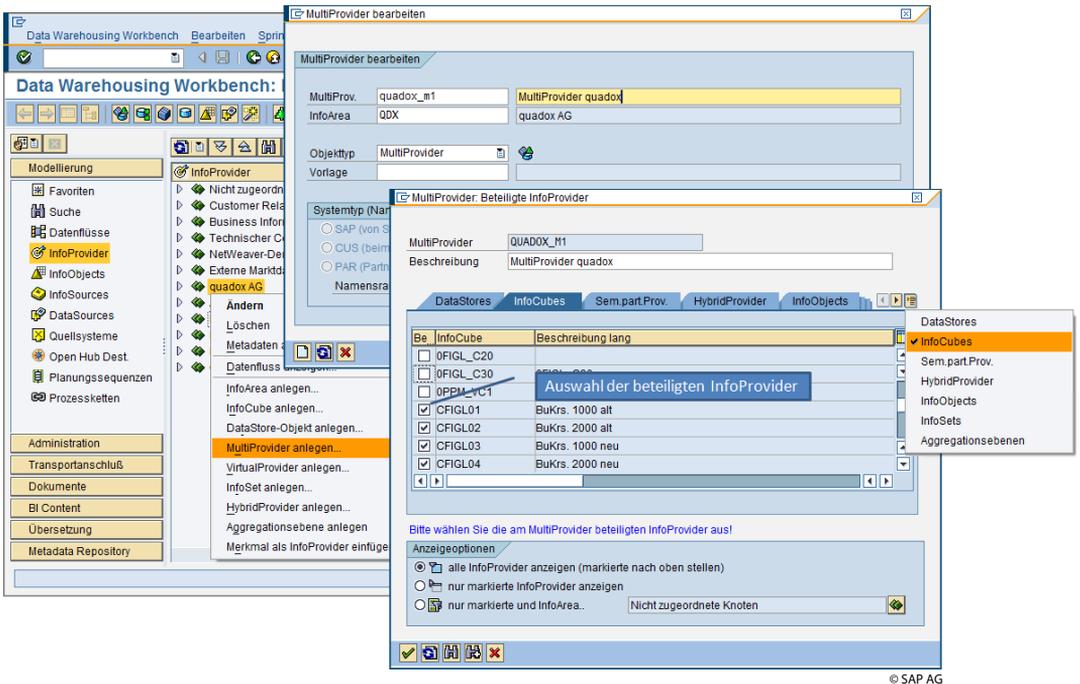
Über die Zusammenführung unterschiedlicher InfoProvider hinaus bilden MultiProvider auch eine logische Schicht über die ausgewählten InfoProvider und abstrahieren damit die Definition der Datenanalyse von den genutzten InfoProvidern. Setzen Sie MultiProvider daher stets ein – selbst wenn nur Daten aus einem einzigen InfoProvider gelesen werden müssen. Weitere Einzelheiten werden im Abschnitt zum BW-Design in Kapitel 35 erläutert.

Welche InfoProvider einem MultiProvider zugrunde liegen sollen, wird bereits beim Anlegen des MultiProviders bestimmt, kann jedoch nachträglich verändert werden. Zur Auswahl stehen dabei alle Typen von *InfoCubes* und *DataStore-Objekten*, *HybridProvider*, *InfoSets* und *Aggregationsebenen* der integrierten Planung – nicht jedoch andere MultiProvider (siehe Abb. 27–1). Werden InfoCubes oder DataStore-Objekte als *semantisch partitionierte Objekte* angelegt, so können entweder die dadurch definierten Partitionen oder das semantisch parti-

1. Siehe Abschnitt 12.4.

tionierte Objekt selbst in den MultiProvider aufgenommen werden. Letzteres ist zu empfehlen, da hierdurch auch die für semantisch partitionierte Objekte spezifischen Optimierungen Anwendung finden². InfoObjekte können nur dann in MultiProvider aufgenommen werden, wenn sie als InfoProvider definiert wurden³.

Abb. 27-1
Anlegen von
MultiProvidern



© SAP AG

Die Zusammenführung mehrerer InfoProvider zu einem MultiProvider erfolgt nach dem Prinzip einer *Union-Abfrage*, in der sämtliche Tabellenfelder (=InfoObjekte) mit identischem Namen aufeinander abgebildet werden. Kennzahlenfelder mit identischem Namen werden dabei aggregiert (siehe Abb. 27-2).

Die InfoObjekte, die ein MultiProvider zur Datenanalyse bereitstellen kann, werden durch die zugrunde liegenden InfoProvider vorgegeben. Diese sind bei der Modellierung ebenso wie bei InfoCubes in Dimensionen zu gliedern, wobei die Gliederung in diesem Fall keine Auswirkung auf die physische Datenhaltung hat, sondern lediglich semantischer Natur ist und nach fachlichen Gesichtspunkten erfolgen kann.

*Identifikation von
Merkmalen*

2. Siehe Abschnitt 27.2.
3. Siehe Kapitel 4.

Abb. 27-2
Zusammenführung von
Daten durch MultiProvider



Auf welche Weise Merkmals-InfoObjekte der einzelnen InfoProvider in einem MultiProvider aufeinander abgebildet werden, ist in der Definition des jeweiligen MultiProviders vorzugeben. Dabei kann für jeden InfoProvider festgelegt werden, ob er Werte zu einem Merkmals-InfoObjekt liefert und aus welchem InfoObjekt der Wert geliefert werden soll (sofern es sich um Merkmale handelt, die auf identische Basismerkmale referenzieren⁴). Die InfoObjekte sind hierfür in der Pflege des MultiProviders entsprechend zu *identifizieren*. Abbildung 27-3 zeigt ein Beispiel, in dem das InfoObjekt OSOLD_TO⁵ im MultiProvider aus den InfoObjekten OSOLD_TO (InfoProvider QDX1) und OCUSTOMER (InfoProvider QDX2) versorgt wird und der InfoProvider QDX3 keinen Wert beisteuert, obwohl er über ein geeignetes InfoObjekt verfügen würde.

4. Siehe Abschnitt 4.1.7.

5. Das InfoObjekt OSOLD_TO referenziert im BI Content auf das InfoObjekt OCUSTOMER.

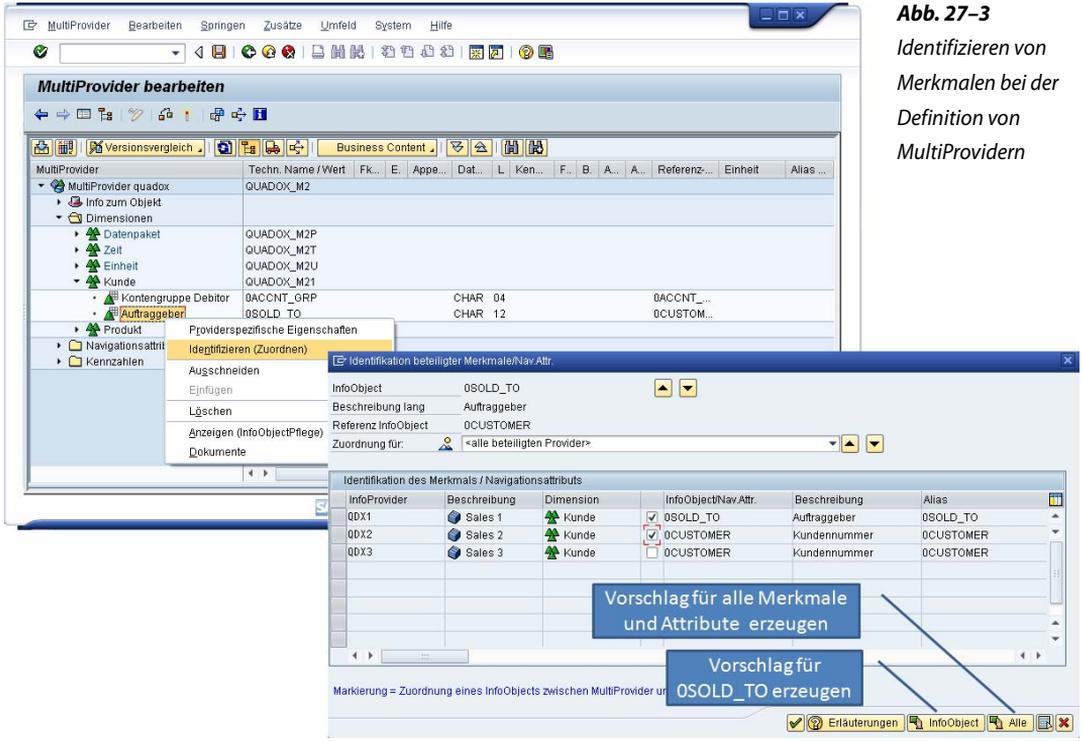


Abb. 27-3
Identifizieren von
Merkmale bei der
Definition von
MultiProvidern

In der Praxis werden üblicherweise immer identische InfoObjekte aufeinander abgebildet. Zu diesem Zweck besteht die Möglichkeit, bei der Identifikation eines Merkmals oder Navigationsattributs einen entsprechenden Vorschlag erzeugen zu lassen. Wird die Identifikation eines Merkmals aus einem InfoProvider nicht aktiviert oder enthält ein InfoProvider kein passendes Merkmal, so wird dem entsprechenden InfoProvider die Lieferung des Initialwerts (#) unterstellt.

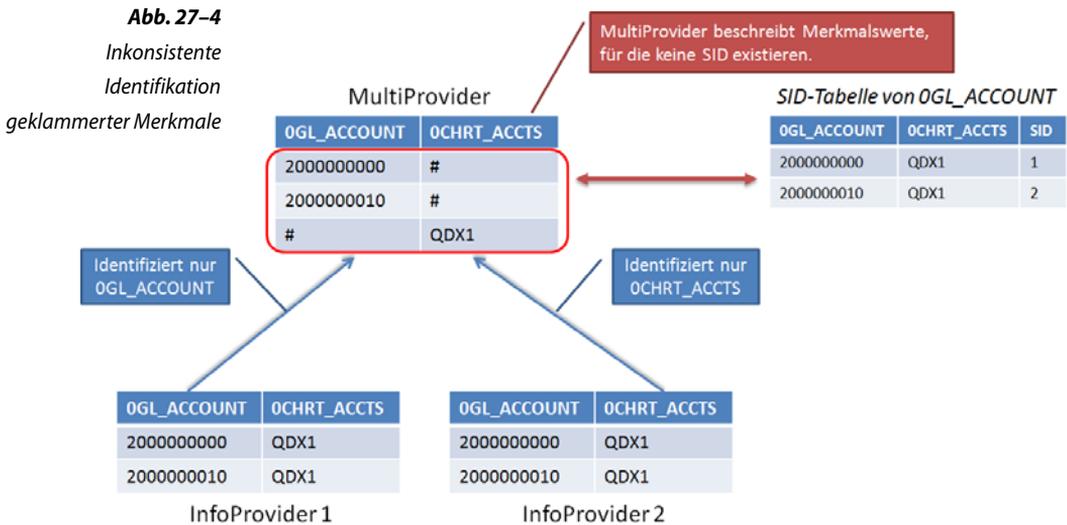
Besondere Beachtung erfordert die Identifikation geklammerter Merkmale und ihrer Klammerungsbestandteile: Diese sollten stets konsistent zueinander identifiziert werden, d.h., jeder InfoProvider sollte das geklammerte Merkmal und alle Klammerungsbestandteile vollständig aus derselben Basis oder gar nicht liefern. So sollten beispielsweise die Merkmale 0GL_ACCOUNT⁶ und 0CHRT_ACCTS beide aus den InfoObjekten eines InfoProviders oder aus den Navigationsattributen desselben InfoObjekts dieses InfoProviders identifiziert werden.

Alternativ können auch beide InfoObjekte nicht identifiziert werden, jedoch würde beispielsweise eine unvollständige Identifikation

*Identifikation
geklammerter Merkmale
und Klammerungs-
bestandteile*

6. Das InfoObjekt 0GL_ACCOUNT (Sachkonto) ist im BI Content an das InfoObjekt 0CHRT_ACCTS (Kontenplan) geklammert.

dazu führen, dass durch den MultiProvider Merkmalswerte geliefert werden, die tatsächlich in keinem der beteiligten InfoProvider vorhanden sind. Abbildung 27–4 verdeutlicht dies anhand zweier InfoProvider, die das InfoObjekt OGL_ACCOUNT mit der Klammerung an OCHRT_ACCTS jeweils nur zum Teil identifizieren.



Ähnlich problematische Situationen können entstehen, wenn Klammerungsbestandteile nicht aus derselben Basis gezogen werden, wenn also beispielsweise OGL_ACCOUNT aus einem Merkmal in den Dimensionen eines InfoCubes und OCHRT_ACCTS aus den Navigationsattributen eines anderen Merkmals des InfoCubes identifiziert wird.

Die Nutzung von Merkmalswerten in der Datenanalyse, zu denen keine SID bestehen, ist für das BW problematisch: Wird eine entsprechende MultiProvider-Abfrage beispielsweise an den BWA gerichtet, verweigert dieser Teile der Abfrage und gibt sie an die Analytical Engine des BW zurück (die derartige Umstände beherrscht). Dies ist zwar in erster Linie ein Performance-Problem, jedoch können auch hier unter Umständen fehlerhafte Ergebnisse entstehen.

Aus diesem Grund meldet das BW die Definition eines MultiProviders grundsätzlich als fehlerhaft, wenn die Identifikation geklammerter Merkmale und ihrer Klammerungsbestandteile so gestaltet ist, dass die beschriebenen Inkonsistenzen entstehen können. Im Sprachgebrauch des BW wird in einem solchen Fall auch von einem Compounding oder *CMP-Problem* gesprochen.

Soll eine derart inkonsistente Modellierung dennoch durchgeführt werden, so kann die Fehlermeldung individuell für jeden MultiProvider in eine Warnung umgewandelt werden (siehe Abb. 27–5).

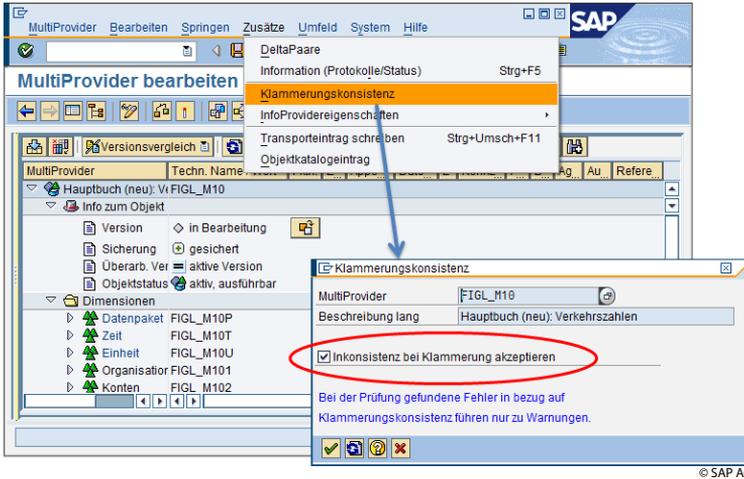


Abb. 27-5
Inkonsistenz bei
Klammerung akzeptieren

Unabhängig davon, ob eine inkonsistente Klammerung zugelassen ist oder nicht, liefert das ABAP-Programm RSCOMPCONS eine Übersicht über MultiProvider mit inkonsistent identifizierten Merkmalen.

Im Falle von Kennzahlen gibt es zwar keine den Merkmalen vergleichbare Identifikation, jedoch kann im Rahmen einer *Selektion* vorgegeben werden, welche Kennzahlen eines InfoProviders im MultiProvider bereitgestellt werden sollen (siehe Abb. 27-6).

Selektion von Kennzahlen

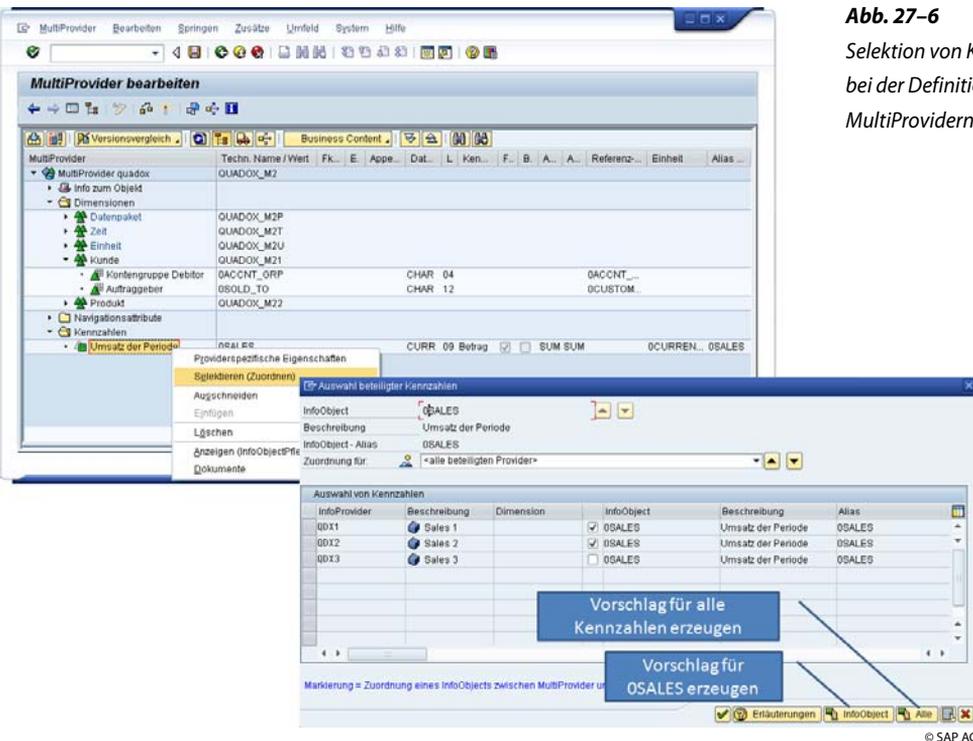


Abb. 27-6
Selektion von Kennzahlen
bei der Definition von
MultiProvidern

Werden identische Kennzahlen mehrerer InfoProvider in einen gemeinsamen Kontext gestellt, so werden ihre Werte durch den MultiProvider summiert. Kennzahlen, die in mehreren InfoProvidern durch unterschiedliche InfoObjekte abgebildet werden, führen auch in einem MultiProvider stets zu unterschiedlichen Kennzahlobjekten. Wird beispielsweise aufgrund eines Modellierungsfehlers die Kennzahl Umsatz einmal als InfoObjekt 0SALES und einmal als ZUMSATZ an einen MultiProvider geliefert, so liefert dieser beide Kennzahlen getrennt voneinander.

Optimierung von
MultiProvidern

Wird eine Query ausgeführt, die auf einem MultiProvider basiert, so wird diese Query in mehrere *Sub-Queries* aufgespalten – je eine für jeden zugrunde liegenden InfoProvider. Jede dieser Sub-Queries ist vergleichbar mit einer Query auf den jeweiligen InfoProvider, kann also ihrerseits in weitere *Teilabfragen*⁷ aufgelöst werden.



Die Begriffe Sub-Query und Teilabfrage werden in der Praxis oft nicht eindeutig differenziert. Um im Rahmen dieses Buchs eine Eindeutigkeit zu schaffen, steht der Begriff der »Sub-Query« hier ausschließlich für die Abfrage eines InfoProviders, die aus einem MultiProvider resultiert, während der Begriff »Teilabfrage« die Zergliederung einer Abfrage (bei der es sich um eine Sub-Query handeln kann) auf einen InfoProvider bezeichnet.

Aus Performance-Gründen kann es von Interesse sein, die Ausführung einzelner Sub-Queries durch die Vorgabe *providerspezifischer Konstanten* und *OLAP-Hints* zu verhindern. Hierauf wird in Abschnitt 27.1.1 eingegangen.

Die Ergebnisse der einzelnen Sub-Queries müssen durch die Analytical Engine weiterverarbeitet werden:

- Ergebnisse von Teilabfragen müssen in ein Gesamtergebnis pro InfoProvider überführt werden.
- Gesamtergebnisse der InfoProvider müssen gemäß der MultiProvider-Konfiguration in ein Gesamtergebnis des MultiProviders überführt werden⁸.
- Ausnahmeaggregationen müssen aus dem Gesamtergebnis berechnet werden⁹.
- Einheiten und Währungen müssen umgerechnet werden¹⁰.

7. Siehe Abschnitt 5.3.2.

8. Siehe Abschnitt 27.1.

9. Siehe Abschnitt 4.3.1 und 4.3.2.

10. Siehe Abschnitt B.1 und B.2 im Anhang.

Speziell beim Einsatz von BWA und HANA können diese Aufgaben teilweise durch diese Systeme übernommen werden. Die hierfür erforderliche Konfiguration der *BWA-Operationen* wird in Abschnitt 27.1.2 erläutert.

27.1.1 Providerspezifische Konstanten und OLAP-Hints

Bei der Beschreibung der semantischen Partitionierung von Datenzeilen wurde bereits die Möglichkeit zur *Definition providerspezifischer Konstanten* beschrieben¹¹. Die Analytical Engine kann diese Einstellungen bei der Ausführung einer Query berücksichtigen und alle InfoCubes und DataStore-Objekte vom Zugriff ausschließen, bei denen durch die Definition einer providerspezifischen Konstante klar ist, dass sie keine Daten liefern werden.

*Providerspezifische
Konstanten*

Einen ähnlichen Ansatz wie die Selektion von providerspezifischen Konstanten verfolgen die sogenannten OLAP-Hints, die den Zugriffen auf *relationale InfoCubes* vorbehalten sind. Bei OLAP-Hints wird jedoch nicht statisch festgelegt, für welche Einzelwerte ein InfoCube Daten liefern kann. Stattdessen wird vor der Ausführung einer Sub-Query auf Basis der Dimensionstabellen eines InfoCubes ermittelt, ob dieser entsprechende Inhalte umfasst.

OLAP-Hints

Um also zu ermitteln, ob ein relationaler InfoCube über einen selektierten Merkmalswert verfügt, wird die Dimensionstabelle, in der sich das Merkmal befindet, auf entsprechende Werte überprüft. Die Nutzung von OLAP-Hints beschränkt sich damit zwangsläufig auf Merkmale in normalen Dimensionstabellen. Merkmale in Line-Item-Dimensionen können nicht durch OLAP-Hints genutzt werden.

Ferner sollten OLAP-Hints nur bei besonders kleinen Dimensionstabellen angewandt werden, damit die Überprüfung performant durchgeführt werden kann – denn schließlich erfolgt die Überprüfung von OLAP-Hints *zusätzlich* zur eigentlichen Analyse der Daten. Soll ein Merkmal in einer großen Dimensionstabelle durch OLAP-Hints analysiert werden, so erzeugt das Durchsuchen der Dimensionstabelle vor jeder Auswertung einen vergleichsweise großen hohen Aufwand und verschlechtert die Performance unter Umständen eher, als dass sie verbessert wird¹².

11. Siehe Abschnitt 12.4.1.

12. Es werden stets die Dimensionstabellen des InfoCubes, nicht jedoch die Dimensionstabellen von Aggregaten durchsucht. Die Auswahl geeigneter Aggregate erfolgt erst nach der Berücksichtigung der OLAP-Hints, sodass das Anlegen von Aggregaten keine Vorteile für OLAP-Hints bietet.



Dimensionstabellen können Merkmalswerte enthalten, die nicht mehr in einem InfoCube verwendet werden. Dieses Phänomen tritt vor allem nach der Komprimierung der Faktentabelle mit Nullwert-Eliminierung, nach dem Löschen von Requests und nach selektivem Löschen auf. Trimmen Sie daher in regelmäßigen Abständen die Dimensionstabellen Ihrer InfoCubes, um derartige Einträge in Dimensionstabellen zu entfernen. Das Verfahren hierzu ist in Abschnitt 5.1.2 erläutert.

Aufgrund der Nachteile, die OLAP-Hints mit sich bringen können, muss ihre Nutzung pro MultiProvider und Merkmal explizit aktiviert werden, indem MultiProvider und Merkmal in der Tabelle RRMULTI-PROVHINT eingetragen werden. Zu diesem Zweck kann die Funktion *Einträge erfassen* in der Transaktion SE16 verwendet werden. Abbildung 27-7 zeigt dies beispielhaft an dem MultiProvider QUADOX_C1 und dem darin enthaltenen Merkmal OSALES_OFF.

Abb. 27-7
Konfiguration von
OLAP-Hints

The screenshot shows two overlapping windows from the SAP Data Browser. The top window is titled 'Data Browser: Einstieg' and shows the 'Einträge erfassen (F5)' function with the table name 'RRMULTI-PROVHINT'. The bottom window is titled 'Tabelle RRMULTI-PROVHINT einfügen' and shows the configuration for the table. The 'InfoProvider' field is set to 'QUADOX_C1' and the 'Merkmal' field is set to 'OSALES_OFF'. There is also a 'Zähler' (Counter) field which is currently empty.

© SAP AG

Zusätzlich zu dem Merkmal ist in der Tabelle RRMULTI-PROVHINT ein Zähler vorzugeben, der festlegt, in welcher Reihenfolge Prüfungen durchgeführt werden sollen. Wenn ein Cube bereits bei einem Merkmal ausscheidet, müssen die nachfolgenden nicht mehr geprüft werden. Merkmale in den kleinsten Dimensionen sollten daher möglichst vor Merkmalen in größeren Dimensionen stehen.

Dabei ist zu beachten, dass bereits ein einziger ungeeignet modellierter InfoCube innerhalb eines MultiProviders ausreicht, um die Nutzung von OLAP-Hints für ein Merkmal unvorteilhaft zu gestalten.

27.1.2 BWA-Operationen

Bis zum Release 7.0 des SAP BW liefern alle eingesetzten Datenbanksysteme ausschließlich die Ergebnisse der an sie gerichteten (Sub-)Queries an die Analytical Engine, die ihrerseits die Zusammenführung der Ergebnisse (gemäß der Definition der MultiProvider) übernimmt.

Für das BW-Release 7.3¹³ bringen BWA und HANA eine eigene Analytical Engine mit sich, die die Funktionalität der Analytical Engine des SAP BW zum Teil ebenfalls bietet: Dies betrifft vor allem die Ausnahmeaggregation und (als Voraussetzung dafür) die MultiProvider-Funktionalität. Sollen die Funktionen der Analytical Engine im BW durch die Funktionalität von BWA/HANA ersetzt werden, so kann zwischen unterschiedlichen Stufen gewählt werden:

- keine Operationen im BWA (BWA-Operation 0)
- Einzelzugriff pro InfoProvider (BWA-Operation 2)
- Standard, d.h. MultiProvider Cluster Access (BWA-Operation 3)
- Ausnahmeaggregation (BWA-Operation 6)

Welche BWA-Operationen für die Queries eines MultiProviders ausgeführt werden sollen, ist in den InfoProvider-Eigenschaften des MultiProviders festzulegen, die aus der Pflege des MultiProviders über den Menüpunkt *Zusätze* → *InfoProvidereigenschaften* → *Ändern* geändert werden können (siehe Abb. 27–8).

Die im MultiProvider hinterlegten Eigenschaften werden beim Anlegen einer Query in diese übernommen. Werden die Eigenschaften eines MultiProviders nachträglich verändert, so hat dies keine Auswirkungen auf die Eigenschaften bestehender Queries.

Bei Bedarf können die Eigenschaften einzelner Queries individuell angepasst werden. Dies erfolgt in der Transaktion RSRT im Menüpunkt *Query* → *Eigenschaften*. Sollen die Eigenschaften mehrerer Queries angepasst werden, so kann dies in der Transaktion RSRT über den Menüpunkt *Umfeld* → *Query Massenflege* geschehen.

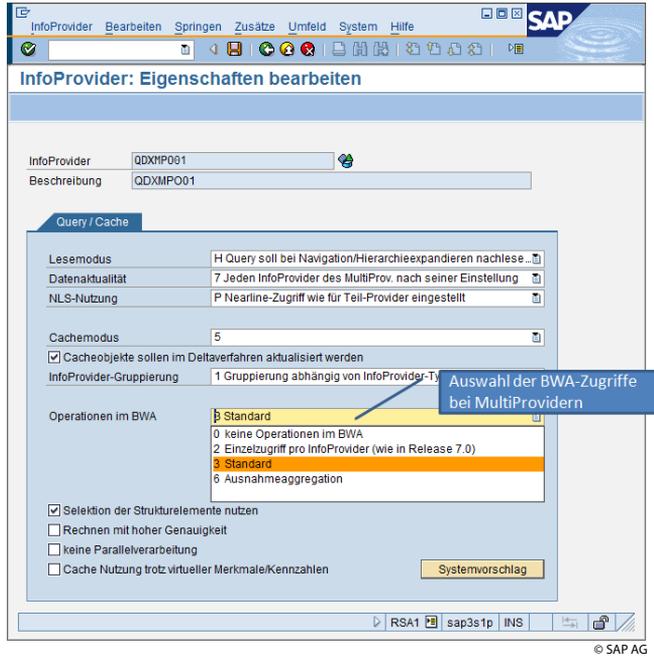
Per Default führt der BWA den sogenannten *MultiProvider Cluster Access* durch. Hierbei übernimmt der BWA die Zusammenführung der an einem MultiProvider beteiligten InfoCubes in Gruppen (Cluster). Ein Cluster definiert sich dabei jeweils durch ein Set von Kennzahlen, die in der Definition des MultiProviders identisch selektiert sind¹⁴. Die Ergebnisse der einzelnen Cluster werden an das BW geliefert, wo die Analytical Engine des SAP BW den Rest der MultiProvider-Zusam-

*MultiProvider Cluster
Access*

13. Ab SP 05.

14. Siehe Seite 531 (Selektion von Kennzahlen).

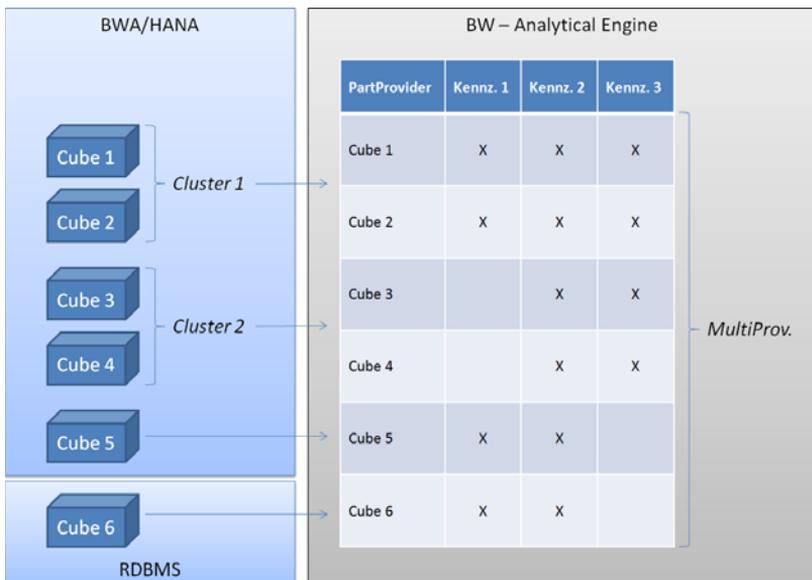
Abb. 27-8
BWA-Operationen eines
MultiProviders festlegen



menführung übernimmt und ggf. auch die Ergebnisse nicht indizierter InfoProvider berücksichtigt.

Abbildung 27-9 skizziert dieses Vorgehen anhand eines MultiProviders mit sechs InfoCubes, von denen fünf im BWA indiziert sind, wobei die Cubes 1 und 2 sowie die Cubes 3 und 4 jeweils über identisch selektierte Kennzahlen verfügen.

Abb. 27-9
MultiProvider Cluster
Access im BWA



Der Cluster Access kann den Datentransfer zwischen BWA und SAP BW reduzieren und verschiebt Rechen- und Speicherlast vom SAP BW in den BWA. Dies kann sich bei einem ausreichend leistungsstarken BWA positiv auf die Gesamtperformance auswirken; bei knapp bemessener BWA-Hardware kann dies den BWA jedoch auch übermäßig stark belasten oder gar überlasten, sodass sich ein Single PartProvider Access (s.u.) unter Umständen günstiger darstellt.

An die Zusammenstellung des MultiProvider-Ergebnisses schließt sich die Durchführung von Ausnahmeaggregationen¹⁵ an, d. h., Kennzahlen mit Ausnahmeaggregationen können erst errechnet werden, nachdem das Ergebnis der MultiProvider-Union vollständig vorliegt und Währungen und Einheiten umgerechnet wurden.

Ausnahmeaggregation

Da Ausnahmeaggregationen die Hinzunahme der jeweiligen Bezugsmerkmale erfordern und damit unter Umständen ein relativ großes Datenvolumen benötigen, um eine relativ kleine Ergebnismenge zu berechnen¹⁶, ist auch die Verlagerung der Ausnahmeaggregation in den BWA/HANA von Interesse, um den Datentransfer zwischen BWA/HANA und SAP BW zu reduzieren.

Die Durchführung von Ausnahmeaggregationen erfolgt per Default in der Analytical Engine des SAP BW, kann aber durch die BWA-Operation 6 (Exception Aggregation) in BWA/HANA verlagert werden. Als Voraussetzung muss der MultiProvider zunächst so konfiguriert sein, dass durch den MultiProvider Cluster Access (s.o.) *alle InfoProvider in einem einzigen Cluster* gelesen werden können.

Dies impliziert, dass die Daten aller InfoProvider im BWA indiziert sein müssen, die zur Berechnung einer Ausnahmeaggregation gelesen werden müssen. Im Falle der HANA-Datenbank müssen alle InfoCubes als HANA-optimierte InfoCubes angelegt sein. Im Falle des BWA werden die noch *fehlenden Cubes im BWA zur Query-Laufzeit indiziert*. Diese temporären Indizes werden automatisch erstellt und wieder gelöscht und enthalten nicht den gesamten Inhalt des jeweiligen InfoCubes, sondern nur die selektierten Merkmale und Kennzahlen. Bei der Indizierung wird auf eventuell geeignete Aggregate¹⁷ zurückgegriffen, wodurch die Performance der Indizierung verbessert wird.

Durch den automatischen Aufbau temporärer Indizes benötigt das System nicht nur Zeit, um die Daten solcher Cubes aus dem relationalen Datenbanksystem zu lesen, sondern auch um die gelesenen Daten

15. Siehe Abschnitt 4.3.1 und 4.3.2.

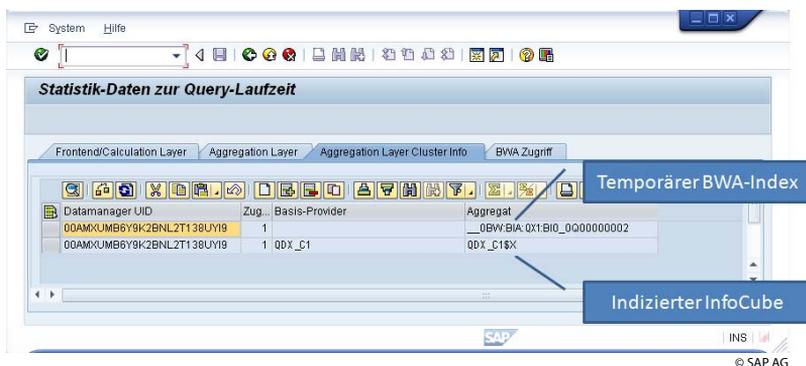
16. Beispielsweise erfordert die Berechnung des durchschnittlichen Umsatzes pro Kunde zunächst den Umsatz aller Kunden, selbst wenn im Ergebnis der Query nur eine Zeile mit dem durchschnittlichen Kundenumsatz in einem Monat abgefragt wird.

17. Siehe Abschnitt 5.3.

in den BWA zu transferieren, zu schreiben und anschließend wieder zu lesen. Dies bedeutet, dass die BWA-Operation 6 nur dann vorteilhaft sein kann, wenn der Performance-Vorteil durch das Lesen der indizierten InfoCubes so ausschlaggebend ist, dass die Performance-Einbußen für die nicht indizierten InfoCubes nicht ins Gewicht fallen.

Ob temporäre Indizes angelegt werden, kann in den Runtime-Statistiken der Analytical Engine¹⁸ kontrolliert werden: Sind in der Cluster-Info Aggregat-Zugriffe ohne Angabe eines Basisproviders zu finden, so wurde das entsprechende Aggregat (=BWA-Index) temporär zur Durchführung der Ausnahmeaggregation angelegt (siehe Abb. 27–10).

Abb. 27–10
Einsatz temporärer Indizes
bei BWA-Operation 6
(Ausnahmeaggregation)



Ob der BWA eine Ausnahmeaggregation berechnen kann, hängt neben der MultiProvider-Konfiguration auch von den jeweils zu berechnenden Kennzahlen und der Art der Ausnahmeaggregation ab. In den folgenden Fällen wird die Ausnahmeaggregation für eine Kennzahl trotz BWA-Operation 6 nicht im BWA, sondern durch die Analytical Engine des BW berechnet:

- Nutzung zeitabhängiger Hierarchien
- Nutzung solcher geklammerter Merkmale und Klammerungsbestandteile, die inkonsistent identifiziert sind¹⁹
- Nutzung virtueller Merkmale oder virtueller Kennzahlen
- Nutzung von Kennzahlen mit Binnenumsatzeliminierung
- Kennzahlen mit variabler Währung (diese Einschränkung ist ab BW7.30 SP5 aufgehoben)
- Kennzahlen mit variabler Einheit
- Query-Formeln, die vor der Aggregation zu berechnen sind
- Nutzung anderer Ausnahmeaggregationen als Durchschnittsbildung, also beispielsweise Bestandsveränderungen, Varianz, erster/letzter Wert und Zählung von Werten

18. Siehe Kapitel 30.

19. Siehe Abbildung 27–4.

Vor allem die zuletzt aufgeführte Einschränkung stellt eine Momentaufnahme dar, die zum Zeitpunkt der Drucklegung gültig ist. Mit der fortlaufenden Entwicklung von BWA und HANA ist damit zu rechnen, dass die Einschränkungen schrittweise reduziert werden.

Durch die zahlreichen Einschränkungen besteht bei der BWA-Operation 6 das Risiko, dass der BWA zwar einen kleinen Teil der zu berechnenden Ausnahmeaggregationen durchführen kann, der größere Teil der Berechnungen jedoch beim BW verbleibt und ein unvermindert großer Datentransfer erfolgen muss. In diesem Fall wird der Ressourcenverbrauch im BW durch den BWA nicht vermindert, sondern nur durch zusätzlichen Ressourcenverbrauch im BWA ergänzt. Verwenden Sie die BWA-Operation 6 daher nur, wenn Sie sich der Vorteilhaftigkeit dieser Operation in der Praxis sicher sind.



Die Nutzung der Analytical Engine des BWA/HANA setzt voraus, dass diese über ausreichend freie Speicherressourcen verfügt, um die Zwischenergebnisse der entsprechenden Operationen ebenfalls im Speicher zu halten. Bei leistungsstarken Systemen wird diese Voraussetzung in der Regel erfüllt sein, jedoch kann der zusätzliche Ressourcenbedarf insbesondere in kleinen oder sehr stark genutzten Systemen zu Problemen führen.

*Einzelzugriff pro
InfoProvider*

Aus diesem Grund besteht mit der BWA-Operation 2 die Möglichkeit, BWA und HANA nur als Datenlieferanten zu nutzen und die Zusammenführung der Daten vollständig der Analytical Engine des BW zu überlassen.

In spezifischen Fällen kann es sinnvoll sein, die Nutzung des BWA vollständig zu verbieten – selbst wenn die verwendeten InfoCubes indiziert sind. Dies erfolgt durch die BWA-Operation 0 und ist vor allem dann sinnvoll, wenn sehr große Ergebnismengen abgefragt werden – beispielsweise durch Analyseprozesse, die auf einer Query basieren.

*Keine Operationen
im BWA*

27.2 Semantisch partitionierte Objekte

Werden InfoCubes oder DataStore-Objekte als semantisch partitionierte Objekte definiert, so handelt es sich bei den daraus abgeleiteten Partitionen um physische vorhandene InfoProvider. Jeder einzelne dieser InfoProvider kann eigenständig oder als Teil eines MultiProviders als Grundlage der Datenanalyse genutzt werden. Gleichzeitig tritt auch das semantisch partitionierte Objekt als virtueller InfoProvider auf, der jedoch nur genutzt werden kann, wenn er in einen MultiProvider eingebunden wird.

Query Pruning

Die Funktionsweise eines semantisch partitionierten Objekts bei der Datenanalyse ist vergleichbar mit einem MultiProvider, dient aber ausschließlich zur Zusammenführung der hierdurch definierten Partitionen.

Dabei ist zu beachten, dass semantisch partitionierte Objekte bereits über Informationen zu ihren Partitionen verfügen, die relevant für die Optimierung der Zugriffe sein können: So ist für jede Partition definiert, welche disjunkten Merkmalsausprägungen in ihnen abgelegt sind. Diese Information nutzt die Analytical Engine des BW zum sogenannten Query Pruning, d.h., zur Laufzeit einer Abfrage entscheidet das BW selbstständig, welche Partitionen potenziell Daten für eine Selektion bereitstellen können.

Ein manuelles Vorgeben der Kriterien, wie dies beim MultiProvider durch providerspezifische Konstanten oder OLAP-Hints erforderlich ist, entfällt bei semantisch partitionierten Objekten.

SPO in BWA/HANA

Aus Sicht von BWA und HANA ist entscheidend, dass alle Partitionen eines semantisch partitionierten Objekts strukturidentisch sind und die Felder aller Partitionen unmittelbar aufeinander abgebildet werden können (es existieren keine vergleichbaren Mechanismen zur Identifikation von Merkmalen oder zur Selektion von Kennzahlen, wie sie aus MultiProvidern bekannt sind).

Damit erfüllen semantisch partitionierte Objekte alle Voraussetzungen, um möglichst viel Funktionalität für die Zusammenführung der Sub-Queries in BWA/HANA zu nutzen (Cluster Access und Ausnahmeaggregation). Im Falle des BWA ist zu beachten, dass semantisch partitionierte Objekte auf zweierlei Arten indiziert werden können:

- in Form der Partitionen,
- in Form des semantisch partitionierten Objekts.

Ob ein semantisch partitioniertes Objekt oder dessen Partitionen im BWA indiziert werden, entscheidet sich dadurch, welches der Objekte in der Transaktion RSDDb zur Indizierung vorgegeben wird. Technisch werden in jedem Fall *BWA-Indizes für jede einzelne Partition* angelegt. Bei der Indizierung des semantisch partitionierten Objekts werden jedoch alle zugehörigen Partitionen automatisch indiziert. Werden neue Partitionen angelegt, so wird dies automatisch in der Indizierung nachvollzogen (während neue Partitionen andernfalls manuell indiziert werden müssten).

Neben den administrativen Aspekten ist auch zu beachten, dass im BWA für semantisch partitionierte Objekte ein eigener logischer Index angelegt wird, der alle indizierten Partitionen umfasst. Wird über

BI-Tools direkt auf den BWA zugegriffen, so werden die Tools hiermit in die Lage versetzt, auch ohne Zutun des SAP BW die jeweils korrekte Definition eines semantisch partitionierten Objekts zu berücksichtigen und alle erforderlichen Partitionen zu lesen.

27.3 InfoSets

Durch InfoSets können die Daten unterschiedlicher Datenziele (Info-Cubes, DataStore-Objekte, InfoObjekte mit Stammdaten) durch relationale *Join-Abfragen* miteinander verknüpft werden (anders als durch MultiProvider, die eine *Union-Abfrage* durchführen).

Der Einsatz von InfoSets zielt dabei auf Datenbestände in *relationalen Datenbanksystemen* ab. Beim Einsatz von BWA oder HANA stellt der Einsatz von CompositeProvidern eine bessere Alternative dar²⁰.

Der Begriff des InfoSets ist im BW unglücklich gewählt. Bereits im SAP ERP gibt es ein InfoSet bzw. eine InfoSet-Query, die durch die Transaktionen SQ02 bzw. SQ10 zu definieren sind. Diese haben mit dem InfoSet im SAP BW nichts zu tun und werden im Kontext des BW als Classic InfoSets bezeichnet.



Anlegen von InfoSets

InfoSets können im InfoProvider-Baum der Data Warehousing Workbench oder alternativ über die Transaktion RS1SET definiert werden. Die Modellierung eines InfoSets hat (im Gegensatz zu den übrigen virtuellen InfoProvidern) keine Ähnlichkeit mit der Modellierung von Info-Cubes. Vielmehr werden InfoSets durch die relationale Verbindung der im InfoSet enthaltenen BW-Objekte definiert. Die einzelnen BW-Objekte werden dabei stets in Tabellenform dargestellt – selbst wenn dies nicht ihrem physischen Modell entspricht (siehe Abb. 27–14).

Zur Definition der Join-Bedingung zwischen zwei Tabellen innerhalb eines InfoSets werden gleiche Felder beider Tabellen miteinander verbunden. Hierfür bieten sich vor allem gleiche InfoObjekte bzw. InfoObjekte mit gleichen Basismerkmalen an. Technisch gesehen können bei einer Join-Verbindung jedoch alle InfoObjekte verbunden werden, bei denen Datentyp und Feldlänge dies zulassen.

Per Default werden sämtliche Join-Bedingungen als sogenannter *Inner Join*²¹ realisiert. Dabei werden diejenigen Datensätze aus zwei verknüpften Tabellen zusammengeführt, für die in den verknüpften Feldern entsprechende Werte in beiden Tabellen existieren. Das bedeu-

Inner Join

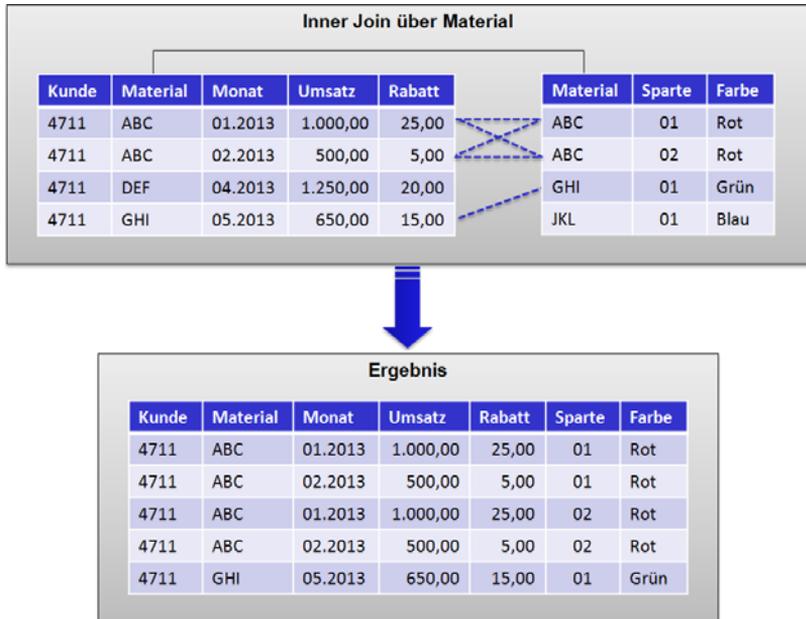
20. Siehe Abschnitt 27.5.

21. Dieser wird in der SAP-Dokumentation auch als Equal Join bezeichnet.

tet auch, dass Datensätze einer der Tabellen, für die kein Pendant in der jeweils anderen Tabelle vorhanden ist, nicht in die Ergebnismenge des Inner Join einfließen (siehe Abb. 27–11).

Abb. 27–11

Inner Join (Equal Join)



*Vervielfachte
Kennzahlwerte*

Bei der Modellierung eines InfoSets muss ein besonderes Augenmerk auf die existierenden Relationen gelegt werden. Wird dies versäumt, so können fehlerhafte Kennzahlwerte daraus resultieren. In Abwandlung zu dem in Abbildung 27–11 gegebenen Beispiel stellt das Material in dem nachfolgenden Beispiel nicht den Schlüssel des Materialstammes dar, sondern nur einen Teil des Schlüssels. Unter einer ansonsten unveränderten Definition des InfoSets ergibt sich das in Abbildung 27–12 dargestellte Ergebnis.

Bei der Modellierung der Relationen in einem InfoSets muss daher stets darauf geachtet werden, in welcher Relation die jeweils verbundenen Felder stehen und welche Auswirkungen dies auf das Ergebnis der Abfrage hat.

Left Outer Join

In speziellen Fällen sollen die Sätze einer Tabelle durch eine Join-Abfrage gelesen werden, selbst wenn keine entsprechenden Datensätze in einer verbundenen Tabelle existieren. Dies ist zum Beispiel dann sinnvoll, wenn transitive Attribute zu einem InfoProvider hinzugelesen werden sollen, jedoch nicht sichergestellt werden kann, dass in jedem Fall entsprechende Stammdaten vorhanden sind.

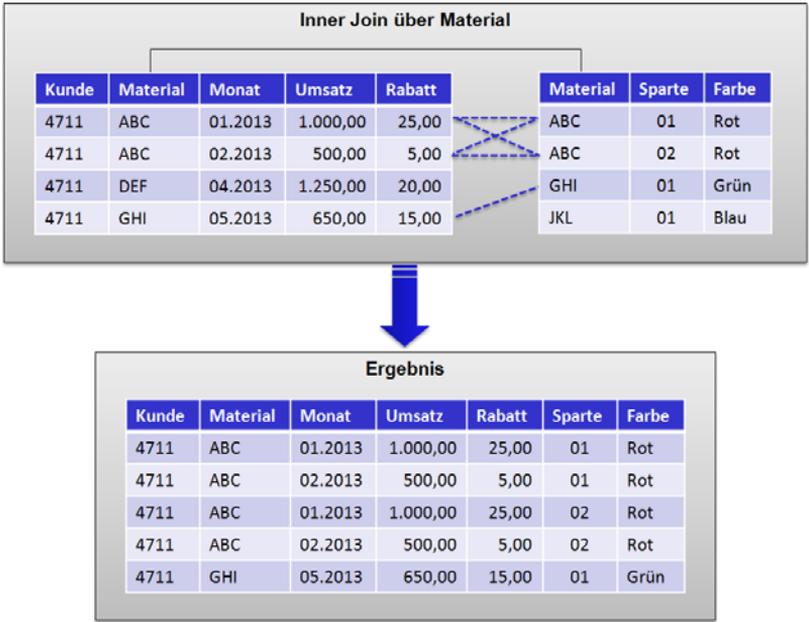


Abb. 27-12
 Vervielfachte
 Kennzahlwerte in InfoSets

Für diese Zwecke kann alternativ zum Inner Join ein sogenannter Left Outer Join definiert werden. Wird eine Tabelle durch einen Left Outer Join mit einer anderen Tabelle verbunden, so werden alle Datensätze aus der einen Tabelle, aber nur die entsprechenden Datensätze aus der anderen Tabelle miteinander verbunden (siehe Abb. 27-13).

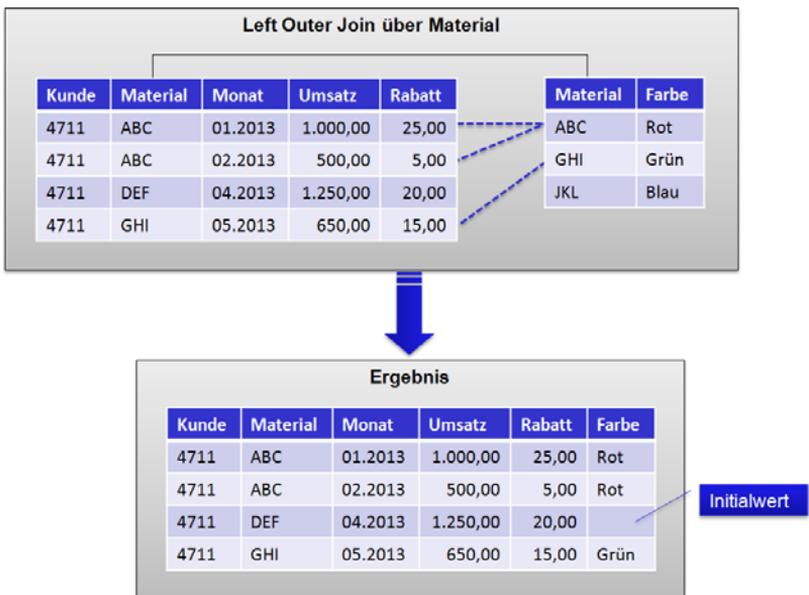


Abb. 27-13
 Left Outer Join

Bei der Definition eines Left Outer Joins ist es damit zwingend erforderlich, festzulegen, aus welcher Tabelle in jedem Fall alle Datensätze gelesen werden sollen und aus welcher Tabelle nur die entsprechend vorhandenen Datensätze oder ansonsten Initialwerte hinzugefügt werden sollen. Um die Bezeichnung »Left Outer« zu Erklärungs Zwecken heranzuziehen, muss also definiert werden, welche der Tabellen auf der »linken« und welche auf der »rechten« Seite in einem Join stehen soll.



Bevor Sie bei der Definition eines InfoSets einen Left Outer Join verwenden, sollten Sie überlegen, ob dieser tatsächlich erforderlich ist oder ob – ggf. durch das Erfüllen weiterer Rahmenbedingungen – alternativ auch ein Inner Join zu nutzen wäre. Left Outer Joins bieten eine schlechtere Performance als Inner Joins und sollten daher gemieden werden.

Das SAP BW kennzeichnet die Definition von Left Outer Joins bei der Pflege von InfoSets, indem der jeweils rechte Operand (also die Tabelle, für die Initialwerte gebildet werden, wenn keine der Join-Bedingung entsprechenden Datensätze gefunden werden) in Weiß abgebildet wird (während Tabellen sonst üblicherweise in Blau dargestellt werden). In Abbildung 27–14 wird ein entsprechendes InfoSet dargestellt.

Abb. 27–14
Definition des Left Outer
Join in InfoSets

The screenshot shows the SAP BW configuration interface for an InfoSet. The main window displays the 'InfoSet Demo der quadox AG (QD_DX_I01) ändern'. On the left, a list of technical names and descriptions is shown. On the right, a table lists the characteristics (Merkmale) for the 'Kunde (0D_CUSTOMER)' table. A 'Left Outer Join' is indicated between the 'Kunde' table and the 'Land (0D_COUNTRY)' table. A context menu is open over the 'Land' table, showing the 'Join Typ' dropdown menu with options: 'Inner Join', 'Left outer Join' (highlighted), and 'Anti Join'. Other options in the menu include 'Alle Felder markieren', 'Alle Felder entmarkieren', 'Attribute neu anordnen', and 'Objekt löschen'.

Technischer Name	Beschreibung	Stichtag
0D_DX_C011	Kunde	
0D_CUSTOMER	Kunde	
0D_DBAREACD	Gebietscode	
0D_DBREGOOD	Regionalcode	
0D_DBASIC1	Branche DB SIC	
0D_DX_C012	Produkt	
0D_PH1	Produktgruppe	
0D_PH2	Produkt	
0D_DX_C013	Vertriebskanal	

Technischer Name	Beschreibung	Stichtag
0D_CUSTOMER	Kunde	
0D_COUNTRY	Land	
0D_INDUSTRY	Branchenschlüssel	
0LONGITUDE	0LONGITUDE	
0LATITUDE	0LATITUDE	
0ALTITUDE	0ALTITUDE	
0SRCID	0SRCID	
0PRECISID	0PRECISID	

Technischer Name	Beschreibung	Stichtag
0D_COUNTRY	Land	
QRESPCNTR	Globale Einheit	

Eine besondere Bedeutung kommt der Verwendung von Filterbedingungen auf Felder der »rechten« Tabelle eines Left Outer Join zu. Bezogen auf Abbildung 27–13 läge eine solche Filterbedingung beispielsweise vor, wenn das Feld »Farbe« auf den Wert »Rot« eingeschränkt wird.

Filterwerte auf
Left-Outer-Tabellen

Je nachdem, ob der Join zuerst und im Anschluss die Filterbedingung (in der WHERE-Klausel) oder die Bedingung als Bestandteil des Joins (in der ON-Klausel) ausgeführt werden, ergeben sich zwei unterschiedliche Ergebnisse, die in Abbildung 27–15 dargestellt werden.

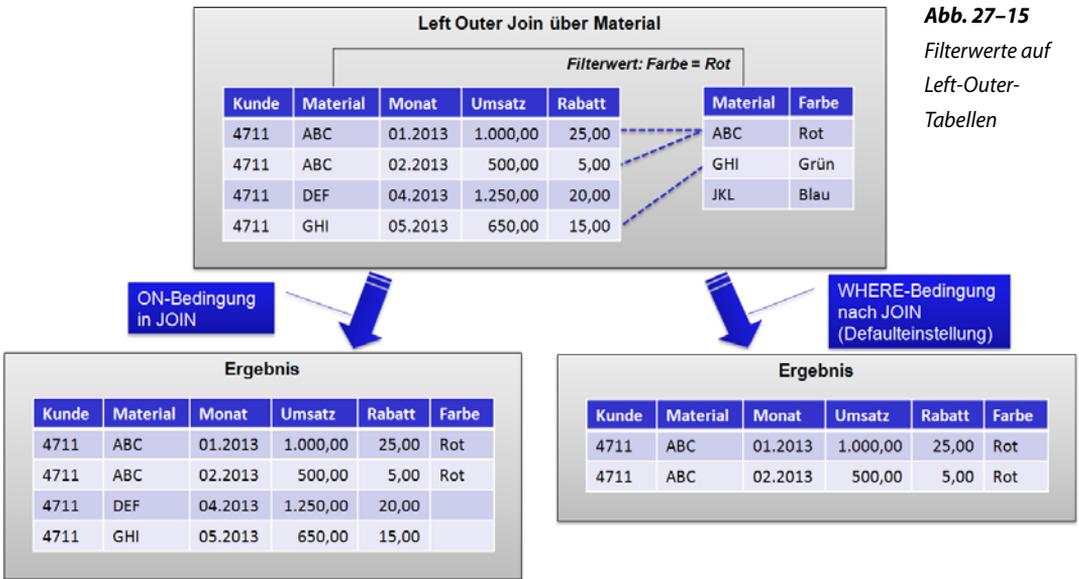
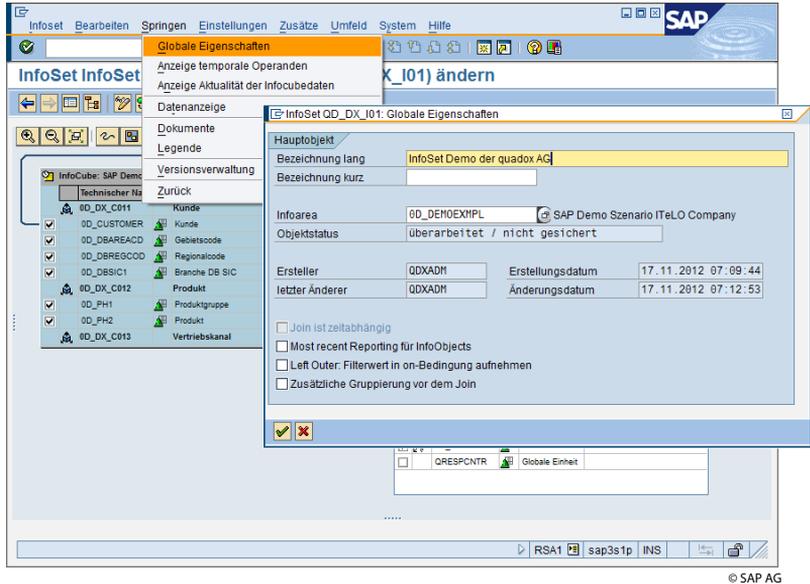


Abb. 27–15
Filterwerte auf
Left-Outer-
Tabellen

Per Default wird die Filterbedingung eines InfoSets in der Where-Bedingung des SQL-Statements integriert, die im Anschluss an den Join ausgeführt wird. Soll dieses Verhalten dahingehend geändert werden, dass die Filterbedingung in die ON-Klausel des SQL-Statements und damit vor dem Join ausgeführt wird, so kann dies durch Aktivieren der Option *Left Outer: Filterwert in on-Bedingung aufnehmen* in den globalen Eigenschaften des InfoSets festgelegt werden (siehe Abb. 27–16).

Diese Einstellung gilt für alle Left Outer Joins des InfoSets und lässt sich nicht auf einzelne Joins innerhalb des InfoSets beschränken.

Abb. 27-16
Globale Eigenschaften
von InfoSets



*Einschränkungen des
Left Outer Join*

Bei der Verwendung eines Left Outer Join ist der rechte Operand an folgende Rahmenbedingungen gebunden:

- Rechte Operanden dürfen nicht mit weiteren InfoProvidern verbunden werden, bilden also – bildlich gesprochen – immer das Ende einer Join-Verbindung.
- InfoCubes stehen aus Performance-Gründen nicht als rechte Operanden zur Verfügung.

*Physischer Zugriff auf
Datenziele*

Der Zugriff auf Datenbestände wird nicht vollständig durch das Datenbanksystem ausgeführt, sondern auch durch die Analytical Engine bearbeitet. Damit ergibt sich eine zweistufige Abfrage in InfoSets: Zuerst stellt das Datenbanksystem eine Grundgesamtheit der Daten zusammen, danach schränkt die Analytical Engine diese Daten weiter ein oder formt sie um.

Dabei ist der Zugriff auf einzelne BW-Objekte für InfoSets spezifisch geregelt und entspricht nicht den Festlegungen, die ansonsten für den Zugriff auf physische Datenziele gelten²². In den nachfolgenden Abschnitten 27.3.1 bis 27.3.3 wird erläutert, wie der physische Zugriff auf InfoObjekte, DataStore-Objekte und InfoCubes durch InfoSets gestaltet werden kann.

22. Siehe Kapitel 28.

27.3.1 InfoObjekte in InfoSets

Wird ein InfoObjekt in einem InfoSet verwendet, so stehen alle Attribute des InfoObjekts für das InfoSet zur Verfügung und können durch relationale Verknüpfung flexibel um die Stammdaten anderer Info-Objekte angereichert werden – und zwar unabhängig davon, ob es sich um Navigations- oder Anzeigeattribute handelt²³.

Auf diese Weise können Stammdatenattribute auch *transitiv* genutzt werden. Das heißt, es können nicht nur die Attribute eines Merkmals, sondern auch die Attribute eines Attributs in eine Abfrage einbezogen werden, indem die Verbindungen zwischen den einzelnen Stammdatentabellen entsprechend im InfoSet definiert werden.

Transitive Attribute

Im Falle von reinen Anzeigeattributen ist nicht sichergestellt, dass verwendete Attributwerte tatsächlich in den Stammdaten des jeweiligen Merkmals existieren. Beispielsweise kann die Postleitzahl 4711 in der Stammdatentabelle des Kunden hinterlegt werden, obwohl diese Postleitzahl in den Stammdaten für das InfoObjekt Postleitzahl nicht existiert. Ein Inner Join auf die Stammdatentabelle Postleitzahl kann demnach zu einer ungewollt reduzierten Ergebnismenge führen. Verwenden Sie zur Verknüpfung von Stammdatentabellen für Anzeigeattribute daher möglichst einen Left Outer Join.



Hierarchien von InfoObjekten stehen in InfoSets nicht zur Verfügung, da diese nicht relational, sondern lediglich durch den internen Aufbau der Inklusionstabellen gelesen werden können.

Darüber hinaus bieten InfoSets einen besonderen Umgang mit nicht aktiven Stammdaten. Diese müssen normalerweise zunächst durch den Attributsänderungslauf²⁴ aktiviert werden, bevor ihre Änderungen in der Datenanalyse zur Verfügung stehen.

*Most Recent Reporting
von Stammdaten*

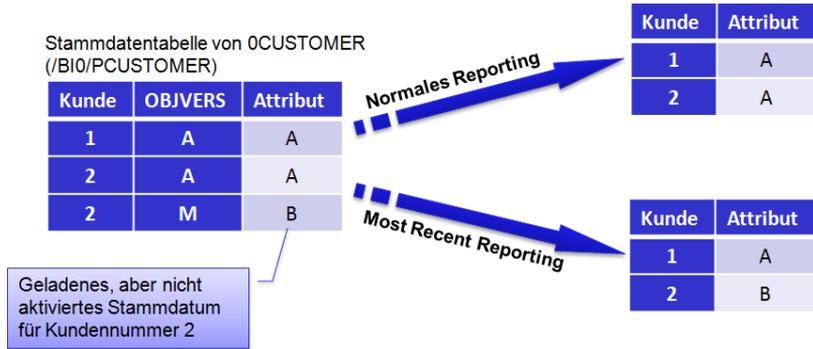
InfoSets hingegen können beim Zugriff auf die Stammdaten von InfoObjekten auch Attribute lesen, die nicht aktiv sind, was in diesem Zusammenhang als *Most Recent Reporting* bezeichnet wird. Abbildung 27-17 stellt den Unterschied zwischen normalem Reporting und Most Recent Reporting von Stammdaten am Beispiel des InfoObjekts OCUSTOMER dar, das über einen Stammdatensatz für den Kunden 2 verfügt, der sowohl in aktiver als auch in neuer (noch nicht aktivierter) Form vorliegt.

Ob das Reporting von Stammdaten durch ein InfoSet in Form von normalem Reporting oder in Form von Most Recent Reporting erfolgen soll, wird in den globalen Eigenschaften des InfoSets über die

23. Gelesen werden ausschließlich die P- und Q-Tabelle eines InfoObjekts.

24. Siehe Abschnitt 21.2.

Abb. 27-17
Most Recent Reporting
von Stammdaten



Option *Most recent Reporting für InfoObjects* festgelegt²⁵ und gilt für alle InfoObjekte innerhalb eines InfoSets. Eine differenzierte Behandlung einzelner InfoObjekte ist nicht möglich.

Temporaler Join

Eine weitere Besonderheit bieten InfoSets mit dem *temporalen Join von zeitabhängigen Stammdatentabellen*. Hierbei wird der jeweils zu verwendende Zeitbezug aus einem Zeitmerkmal des InfoSets abgeleitet. Auf diese Weise ist es möglich, nicht nur einen Stammdatensatz aus einem zeitabhängigen InfoObjekt zu lesen (nach einem vorher definierten Stichtag), sondern pro Datensatz in den Bewegungsdaten einen Zeitbezug zu ermitteln, der beim Lesen der zeitabhängigen Stammdatenattribute Anwendung findet.

Die Merkmale, aus denen der Zeitbezug eines Datensatzes im InfoSet abgeleitet werden soll, werden im Kontextmenü des InfoCubes oder DataStore-Objekts im InfoSet bestimmt (siehe Abb. 27-18). Derart gekennzeichnete Merkmale werden auch als *temporale Operanden* bezeichnet.

Temporale Operanden beschreiben einen Gültigkeitszeitraum, der den Zeitbezug des jeweiligen InfoCubes/DataStore-Objekts beschreibt, der für die Verknüpfung zeitabhängiger Stammdatentabellen herangezogen wird. Der Gültigkeitszeitraum leitet sich dabei entweder aus zwei Stichtagen ab, die Anfang und Ende der Gültigkeit beschreiben, oder aus einem Zeitintervall.

25. Siehe Abbildung 27-16.

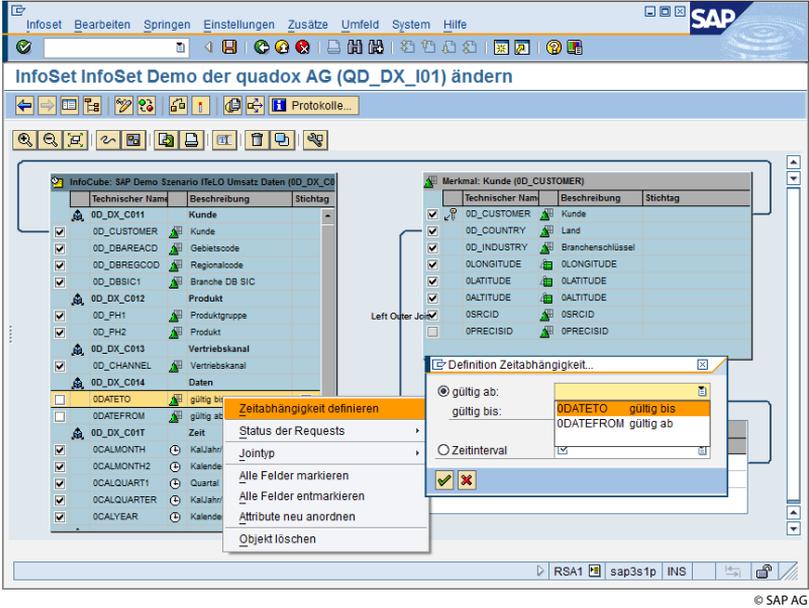


Abb. 27-18
Definition temporaler Operanden

Die beiden *Stichtage* zur Beschreibung des Gültigkeitszeitraums können in Form von Zeitmerkmalen oder Merkmal-InfoObjekten vom Typ Datum angegeben werden. Im Falle von Zeitmerkmalen vom Typ Datum bzw. dem Zeitmerkmal OCALDAY ist der genaue Stichtag, den das angegebene InfoObjekt beschreibt, eindeutig. Werden andere Zeitmerkmale angegeben (z. B. OCALMONTH), so ist zusätzlich anzugeben, ob

- der erste Tag,
- der letzte Tag,
- ein bestimmter Tag (z. B. der dritte Tag)

den genauen Zeitpunkt bestimmt.

Im Falle der Zeitmerkmale OCALWEEK, OCALMONTH, OCALQUARTER, OCALYEAR, OFISCPER und OFISCYEAR können Anfang und Ende des Gültigkeitszeitraums auch direkt aus dem *Zeitintervall* abgeleitet werden, das durch diese Merkmale beschrieben wird. Bei der Angabe des Zeitmerkmals OCALMONTH bilden beispielsweise der erste und letzte Tag des Monats den Gültigkeitszeitraum.

27.3.2 DataStore-Objekte in InfoSets

Bei der Verwendung von DataStore-Objekten in InfoSets wird immer auf die DataStore-Tabelle mit aktiven Daten zugegriffen – und zwar gleichgültig, um welchen Typ von DataStore-Objekt es sich handelt.

Dabei stehen sämtliche Schlüssel- und Datenfelder für die Datenanalyse zur Verfügung, nicht jedoch die Attribute der Merkmale – selbst wenn die SID-Erzeugung eines DataStore-Objekts aktiviert ist. Sollen Stammdatenattribute in die Analyse eingebunden werden, so müssen sie immer durch Modellierung der entsprechenden Join-Bedingungen dem InfoSet beigesteuert werden.

27.3.3 InfoCubes in InfoSets

Seit der Version 7 des SAP BW können auch InfoCubes in die Definition eines InfoSets einbezogen werden. Dabei wird die Aufnahme beliebig vieler InfoCubes in ein InfoSet zwar nicht verhindert – offiziell unterstützt werden jedoch *maximal zwei InfoCubes pro InfoSet*. Aus Performance-Gründen kann ein InfoCube nicht als rechter Operand eines Left Outer Joins definiert werden.

Zur Analyse stehen sämtliche Merkmale und Kennzahlen eines InfoCubes zur Verfügung, nicht jedoch die Attribute der Merkmale – selbst wenn sie als Navigationsattribute definiert sind. Sollen Stammdatenattribute in die Analyse eingebunden werden, so müssen sie immer durch Modellierung der entsprechenden Join-Bedingungen dem InfoSet beigesteuert werden.

Verwendung von
Aggregaten

Existieren zu einem InfoCube Aggregate, die

- in der Query selektierte Kennzahlen des InfoCubes,
- in der Query selektierte Merkmale des InfoCubes oder
- für den Join mit anderen InfoProvidern des InfoSets benötigte Merkmale

enthalten, so liest das BW automatisch Daten aus einem entsprechenden Aggregat. Teilabfragen²⁶ werden bei der Ausführung von InfoSet-Queries nicht gebildet, sodass ein Aggregat stets den vollständigen Informationsbedarf eines Navigationsschritts erfüllen muss.



Innerhalb eines InfoSets ist es nicht möglich, Indizes des BWA zu nutzen. Weichen Sie beim Einsatz des BWA auf die Verwendung von CompositeProvidern aus.

Requeststatus/
Datenaktualität

Analog zum »normalen« Zugriff auf die Daten eines InfoCubes (d.h. außerhalb von InfoSets) wird auch bei einem Zugriff durch InfoSets der Status der einzelnen Requests überprüft²⁷.

26. Siehe Abschnitt 5.3.2.

27. Siehe Kapitel 28.

Anders als beim normalen Zugriff ist der zu verwendende Requeststatus jedoch nicht in der jeweiligen Query zu definieren, sondern im InfoSet selbst (siehe Abb. 27–19).

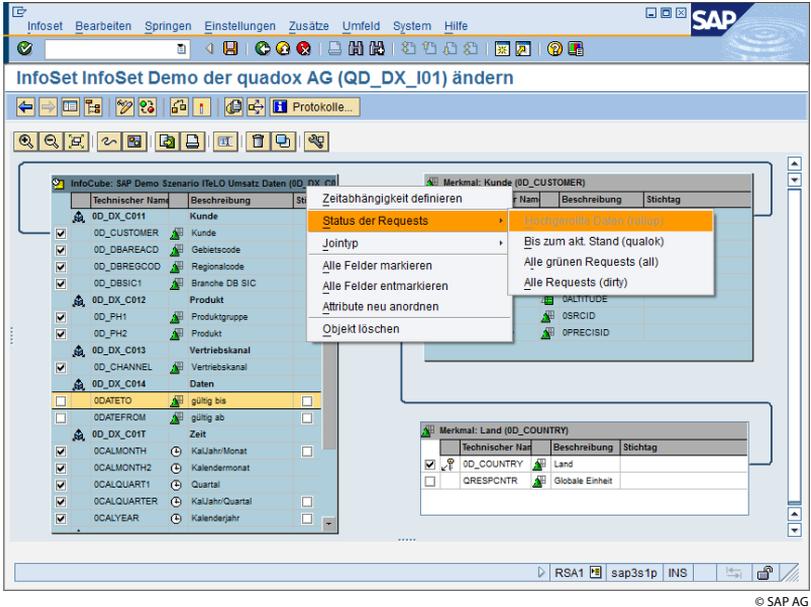


Abb. 27–19
 Behandlung des
 Requeststatus in InfoSets

27.4 TransientProvider

TransientProvider dienen in erster Linie dazu, Daten für die Analyse durch SAP-BI-Tools bereitzustellen, ohne dabei BW-Funktionalität zu verwenden. Die bereitgestellten Datenmodelle werden vielmehr aus Anwendungen abgeleitet, deren Daten in einem TransientProvider abgelegt sind. Bei diesen Anwendungen kann es sich um einen *Analyseprozess* oder um einen *BW Workspace* handeln, die ihre Daten in einem analytischen Index ablegen. Mithilfe der Transaktion `RSDD_LTIP_PUBLISH` können ferner beliebige logische Indizes des BWA als TransientProvider verfügbar gemacht werden. In diesen Fällen dient der Einsatz eines TransientProvider vor allem dem Ziel der Ad-hoc-Analyse.

TransientProvider sind zurzeit nicht in das Transportwesen integriert und müssen daher direkt auf Produktivsystemen angelegt werden (ebenso wie analytische Indizes); sie können auch nicht in MultiProvidern aufgenommen werden. Damit ist der Einsatz von TransientProvidern derzeit keine ernsthafte Option für die Gestaltung von Data-Warehousing-Prozessen. Setzen Sie TransientProvider daher ausschließlich zum Zweck des Ad-hoc-Reporting ein.



Analyse nahe Echtzeit
in SAP ERP

Analytische Indizes können ferner zur Analyse von Daten nahe Echtzeit eingesetzt werden. Sie zielen dabei vor allem auf SAP-ERP-Systeme, die auf Basis der HANA-Datenbank betrieben werden und deren Datenbestände ohne weitere Extraktions- und Aufbereitungsvorgänge oder redundante Datenspeicherung analysiert werden sollen. Die Zusammenstellung der Daten muss dabei entweder durch Analytic Views oder Calculation Views der HANA-Datenbank oder durch InfoSet-Queries geleistet werden.

Analytic Views und Calculation Views müssen im HANA Studio implementiert werden und können mithilfe der Transaktion `RSDD_HM_PUBLISH` als analytischer Index angelegt und als TransientProvider verfügbar gemacht werden (siehe Abb. 27–20).

InfoSet-Queries (nicht zu verwechseln mit Infosets!) können in der Transaktion `SQ02` angelegt werden und mithilfe der Transaktion `SQBWP` als TransientProvider verfügbar gemacht werden. Hierfür ist das Flag in der Spalte *BI-Freigabe* zu aktivieren und eine *InfoArea* anzugeben, in der der TransientProvider ausgewiesen werden soll. Dieser Ausweis erfolgt allerdings nur beim Anlegen von Queries im Query-Designer. In der Data Warehousing Workbench tauchen TransientProvider *nicht* auf.

InfoSet-Queries können selbst dann als TransientProvider definiert werden, wenn keine HANA-Datenbank im Einsatz ist.

Analyse nahe Echtzeit in
SAP BW

Der Einsatz von TransientProvidern im SAP ERP beschränkt die Inhalte der Analyse auf das jeweilige SAP-ERP-System. Die Inhalte weiterer SAP-ERP-Systeme ebenso wie die Daten von Nicht-SAP-Systemen können auf diese Weise nicht in einem gemeinsamen Kontext analysiert werden.

Abhilfe schafft hier die Möglichkeit, Tabelleninhalte per SAP Landscape Transformation (SLT) oder BO Data Services in ein Schema der HANA-Datenbank zu replizieren, auf der auch ein SAP-ERP- oder SAP-BW-System betrieben wird²⁸. Auch diese replizierten Tabellen können mithilfe der Transaktion `RSDD_HM_PUBLISH` als analytischer Index angelegt und als TransientProvider verfügbar gemacht und ausgewertet werden.

Wird die Transformation `RSDD_HM_PUBLISH` in einem BW-System aufgerufen, so können zu den Feldern des angelegten analytischen Index Referenzen auf InfoObjekte hinterlegt werden (siehe Abb. 27–20). Derart referenzierende Felder stehen in TransientProvidern mit ebender Funktionalität bereit, die auch normale InfoObjekte bieten, also beispielsweise der Anzeige von Texten, Attributen oder Hierarchien.

28. Details zur gemeinsamen Nutzung einer HANA-Datenbank durch das BW und weitere Applikationen liefert der Hinweis 1661202 im SAP Support Portal.

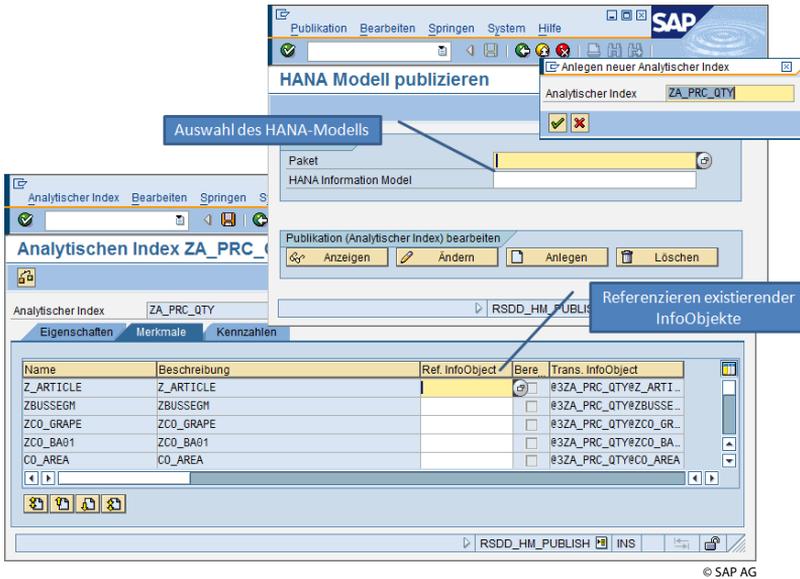


Abb. 27-20
Anlegen von
TransientProvidern für
HANA-Modelle

TransientProvider, die auf Basis von InfoSets angelegt werden, bieten nicht die Möglichkeit, Felder auf InfoObjekte zu referenzieren.

27.5 CompositeProvider

CompositeProvider zielen darauf, die Fähigkeiten von BWA und HANA für das BW verfügbar zu machen. Dabei wird ebenso die Möglichkeit zum Definieren von *Join-Bedingungen* unterstützt wie zum Definieren von *Union-Abfragen*.

CompositeProvider können nicht in MultiProvidern aufgenommen werden und umgekehrt, sodass hier eine Konkurrenzsituation beider InfoProvider entsteht. Die Definition von CompositeProvidern erfolgt dabei mit sehr großen Freiheitsgraden und lässt semantische Prüfungen, die in MultiProvidern obligatorisch sind, unbeachtet (bspw. können InfoObjekte miteinander verknüpft werden, die nicht über dasselbe Basismerkmal verfügen und damit unterschiedliche Stammdaten aufweisen können). Nutzen Sie CompositeProvider daher im BW-Umfeld vorrangig zum Prototyping und für Ad-hoc-Analysen und verwenden Sie für den produktiven Einsatz bevorzugt MultiProvider.



CompositeProvider stellen sich damit als Alternative sowohl zu InfoSets als auch zu MultiProvidern dar, wobei in beiden Fällen spezifische Einschränkungen zu beachten sind. So bieten CompositeProvider keine Funktionen zur Abbildung *temporaler Joins* oder für *Most Recent Reporting*. Ebenso können *keine Aggregationsebenen* für die

integrierte Planung und *keine HybridProvider* in einen CompositeProvider aufgenommen werden.

Beim Anlegen von CompositeProvidern ist zwischen zwei Aufrufen der Modellierungsoberfläche zu unterscheiden: einem, der sich auf die Verwendung von *analytischen Indizes* beschränkt (Transaktion RSLIMO), und einem, der neben analytischen Indizes auch *InfoObjekte*, *InfoCubes*, *DataStore-Objekte* und *semantisch partitionierte Objekte* berücksichtigt (Transaktion RSLIMOBW).

Letztere Modellierungsoberfläche ist Systemen mit HANA-Datenbank vorbehalten; die berücksichtigten DataStore-Objekte und InfoCubes müssen dabei nicht zwangsläufig HANA-optimiert sein. Nutzer des BWA können CompositeProvider ausschließlich auf Basis analytischer Indizes bilden.

Definition von
CompositeProvidern

Die beiden Möglichkeiten zum Aufruf der Modellierungsoberfläche für CompositeProvider spiegeln die beiden möglichen Einsatzgebiete von CompositeProvidern wider: Die Transaktion RSLIMO zielt ausschließlich auf analytische Indizes ab – also auf Datenstrukturen, die vor allem temporärer Natur sind und beispielsweise auch in BW Workspaces definiert werden können. Mithilfe der Transaktion RSLIMO werden CompositeProvider daher unmittelbar in dem System angelegt, auf dem der CompositeProvider auch zum Einsatz kommen soll. Ebenso wie analytische Indizes können auch CompositeProvider, die in der Transaktion RSLIMO definiert werden, nicht durch das Transportwesen transportiert werden²⁹.

Dem gegenüber steht die Transaktion RSLIMOBW, in der auch BW-Objekte zur Analyse kommen können. Enthält ein dort definierter CompositeProvider keine analytischen Indizes, so kann die Definition des CompositeProviders über das Transportwesen transportiert werden.

Bei der Modellierung eines CompositeProviders muss im Kontextmenü für jedes aufgenommene Objekt bestimmt werden, ob dessen Daten über eine Union- oder eine Join-Abfrage in die Ergebnismenge des CompositeProviders eingehen sollen. Lediglich semantisch partitionierte Objekte dürfen ausschließlich durch eine Union-Abfrage in das Ergebnis eingehen.

Die Felder in der Ergebnismenge des CompositeProviders werden per Drag & Drop aus den aufgenommenen Objekten zusammengestellt.

Union

Im Falle einer Union-Abfrage können Objekte beliebige Felder zur Ergebnismenge des CompositeProviders beisteuern. Die Zuordnungen, die per Drag & Drop hergestellt werden, sind dabei vergleichbar mit der Identifikation von Merkmalen, wie sie bei der Definition von

29. Siehe Anhang G.

MultiProvidern durchgeführt werden muss. Dabei gelten jedoch nicht dieselben Restriktionen wie bei MultiProvidern, d.h., es können beliebige Felder in derselben Spalte zusammengeführt werden – unabhängig davon, ob sie auf dasselbe Basismerkmal referenzieren.

Als Besonderheit des CompositeProviders können Konstanten für Objekte hinterlegt werden, die Daten per Union-Abfrage bereitstellen und zu einem bestimmten Feld keinen Wert liefern. Im Gegensatz dazu liefern derartige Objekte in MultiProvidern ausschließlich den Initialwert (#).

Objekte, die ihre Werte über eine Join-Abfrage dem CompositeProvider beisteuern sollen, erhalten die Verknüpfung darüber, dass ein Feld per Drag & Drop mit einem bestehenden Feld der Ergebnismenge im CompositeProvider verbunden wird. Wird ein Feld hingegen neu in den CompositeProvider aufgenommen, so wird dieses über den definierten Join hinzugelesen.

Join

Abbildung 27–21 skizziert die Modellierung eines CompositeProviders.

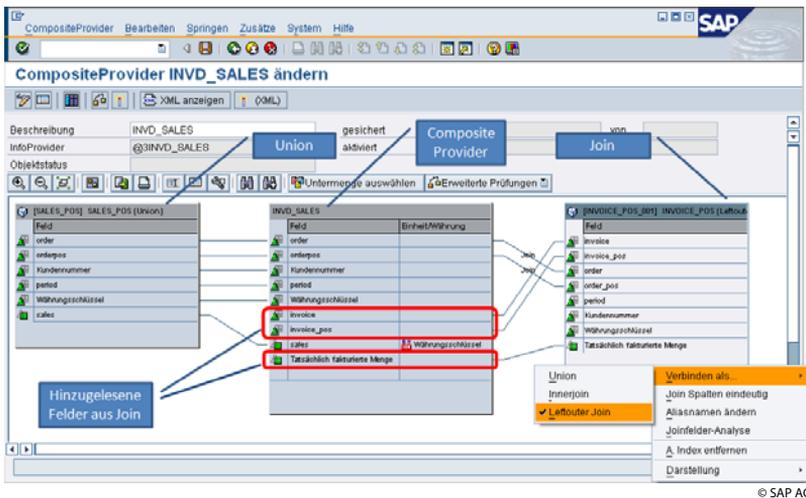


Abb. 27–21
Modellierung von
CompositeProvidern

Aus CompositeProvidern wird seitens BWA/HANA ein Calculation Index generiert, der seinerseits durch einen *TransientProvider* für die Datenanalyse verfügbar gemacht wird (und nicht als Objekt in der Data Warehousing Workbench auftaucht).

Existiert in analytischen Indizes eine Referenz auf bestehende Info-Objekte, so wird diese Referenz auch auf den CompositeProvider übertragen, sodass in Queries auf bestehende Stammdaten und Hierarchien zugegriffen werden kann.

27.6 VirtualProvider

Zur Implementierung spezifischer Analyseanforderungen, die nur durch eigendefinierte Programmlogik zu erfüllen sind, können Virtual-Provider definiert werden, die ihre Daten unabhängig von der Datenbewirtschaftung aus einem selbst entwickelten Programmcoding erhalten.

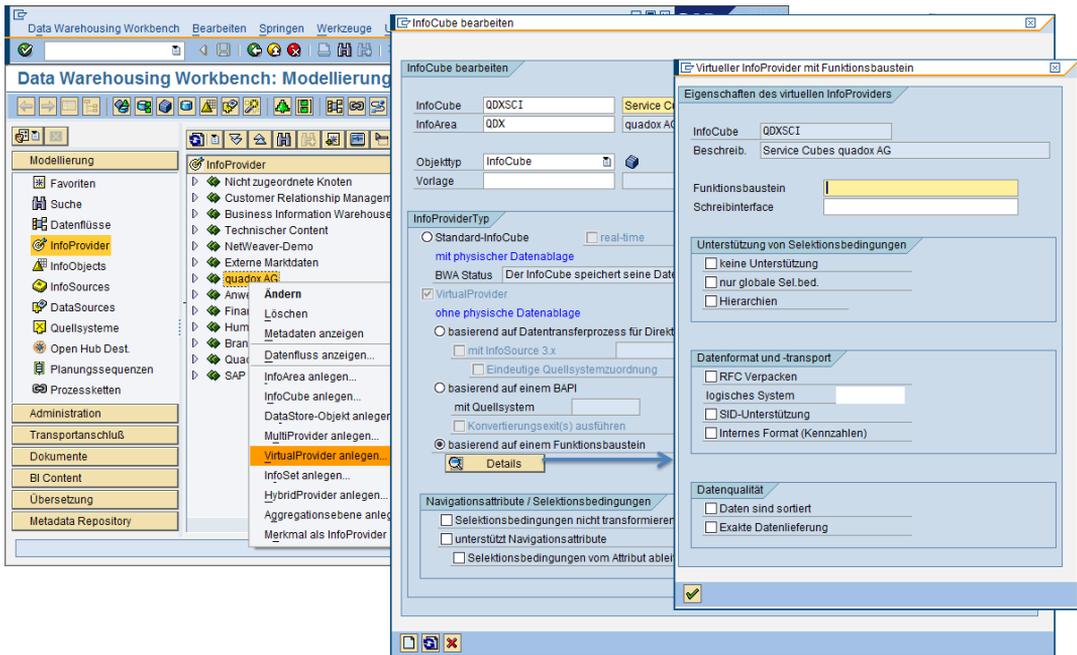
Obwohl ein VirtualProvider mit Funktionsbaustein nur in den Metadaten des SAP BW definiert wird, erfolgt die Definition ebenso wie bei einem InfoCube, d.h., es müssen Dimensionen definiert und mit InfoObjekten versehen werden, Navigationsattribute bestimmt werden usw. Damit präsentieren sich diese Cubes bei der Analyse in gleicher Form wie alle anderen InfoCubes.

Es sind zwei Typen von VirtualProvidern zu unterscheiden:

- VirtualProvider mit Funktionsbaustein, die vor allem zur Implementierung eigener Anforderungen eingesetzt werden können,
- VirtualProvider mit BAPI, die vor allem für den Zugriff auf Daten von Drittanbietern genutzt werden können.

Abb. 27–22
Anlegen eines
VirtualProviders mit
Funktionsbaustein

Welcher Typ von VirtualProvider angelegt werden soll, wird beim Anlegen des VirtualProviders festgelegt. Im Falle eines VirtualProviders mit Funktionsbaustein sind dabei auch Vorgaben zur Gestaltung der Funktionsbaustein-Schnittstelle verbunden (siehe Abb. 27–22).



VirtualProvider mit Funktionsbaustein und BAPI werden nachfolgend in den Abschnitten 27.6.1 und 27.6.2 erläutert. In Abschnitt 27.6.3 wird ergänzend beschrieben, wie Abfragen auf derartige InfoProvider durch BWA und HANA beschleunigt werden können.

VirtualProvider für Direktzugriff werden zur Analyse von Daten nahe Echtzeit eingesetzt und sind in Abschnitt 22.1 beschrieben.

27.6.1 VirtualProvider mit Funktionbaustein

Zur Berücksichtigung spezifischer Anforderungen können VirtualProvider ihre Daten unmittelbar aus einem selbst programmierten Funktionsbaustein beziehen. Dieser ist vor dem Anlegen des VirtualProviders zu implementieren und muss mit den folgenden Schnittstellenparametern versehen werden:

```

IMPORTING
  i_infoprov      type rsinfoprov      "InfoProvider"
  i_th_sfc        type rsdri_th_isfc    "benötigte Merkmale"
  i_th_sfk        type rsdri_th_isfk    "benötigte Kennzahlen"
  i_t_range       type rsdri_t_range    "globale Einschränkungen"
  i_tx_rangetab   type rsdri_tx_rangetab "weitere Einschränkungen"
  i_first_call    type rs_bool          "erster Aufruf des Bausteins"
  i_packagesize   type i               "Paketgröße"
  i_tsx_hier      type rsdri_tsx_hier   "Hierarchie-Einschränkungen"

EXPORTING
  e_t_data        type standard table   "Rückgabetablelle"
  e_end_of_data   type rs_bool          "letztes Datenpaket"
  e_t_msg         type rs_t_msg         "Meldungen an Frontend-Tool"

```

Bei der so vorgegebenen Schnittstelle wird davon ausgegangen, dass

- Selektionen auf Hierarchieknoten in Form der Knoten an den Funktionsbaustein übergeben werden,
- Merkmalsselektionen in Form von Merkmalswerten und nicht in Form von SIDs an den Funktionsbaustein übergeben werden,
- der Funktionsbaustein im BW-System ausgeführt wird und nicht (mithilfe eines RFC) in einem anderen System.

Inhalte und Verwendung der Parameter können sich in Abhängigkeit von den Optionen verändern, die beim Anlegen des VirtualProviders gewählt werden.

Durch die Auswahl der Option *keine Unterstützung* werden keine Selektionsbedingungen der Query an den Funktionsbaustein übergeben. Bei der Ermittlung des Ergebnisses muss sich der Funktionsbaustein ausschließlich an den ausgewählten Merkmalen und Kennzahlen orientieren. Die in den Queries vorgenommenen Selektionen werden

Keine Unterstützung von Selektionsbedingungen

erst durch die Analytical Engine auf den zurückgegebenen Daten prozessiert.



Das Umwandeln der Selektionsbedingungen hat keinen Einfluss auf das Ergebnis einer Query, da die Analytical Engine nach Erhalt des Ergebnisses ebenfalls die Selektion prozessiert und zu viel gelieferte Daten gegebenenfalls selbstständig herausfiltert.

Nur globale
Selektionsbedingungen

Durch die Auswahl der Option *nur globale Sel.bed.* werden nur die globalen Einschränkungen einer Query an den Funktionsbaustein übergeben. Diese umfassen Einschränkungen in Filterwerten, freien Merkmalen und Merkmalen im Aufriss. Weitere Einschränkungen (zum Beispiel eingeschränkte Kennzahlen) werden nicht übergeben.

Selektion von
Hierarchieknoten

Bei der Implementierung des Funktionsbausteins kann man sich entscheiden, die Selektion von Hierarchieknoten zu unterstützen oder nicht. Wird die Selektion auf Hierarchieknoten unterstützt, so werden entsprechende Einschränkungen in Form von Hierarchieknoten an den Funktionsbaustein übergeben. Zu diesem Zweck wird in der Tabelle I_T_RANGE bzw. I_TX_RANGETAB ein Satz mit dem Feld COMPOP = HI erzeugt. Die Zahl, die in diesem Satz im Feld LOW zu finden ist, spezifiziert den Satz in der Tabelle I_TSX_HIER über das Feld POSIT, in dem die entsprechende Hierarchieeinschränkung zu finden ist.

Kann der Funktionsbaustein eine derartige Selektion *nicht* verarbeiten, so kann die Analytical Engine die Einschränkung vor der Übergabe in Form von Merkmalswerten umwandeln. Die Option *Hierarchien* muss in diesem Fall deaktiviert sein. Die Schnittstelle des Funktionsbausteins muss dann folgendermaßen gestaltet sein:

```

IMPORTING
  i_infoprov      type rsinfoprov      "InfoProvider"
  i_th_sfc        type rsdri_th_isfc    "benötigte Merkmale"
  i_th_sfk        type rsdri_th_isfk    "benötigte Kennzahlen"
  i_t_range       type rsdri_t_range    "globale Einschränkungen"
  i_tx_rangetab  type rsdri_tx_rangetab "weitere Einschränkungen"
  i_first_call    type rs_bool         "erster Aufruf des Bausteins"
  i_package_size  type i               "Paketgröße"
EXPORTING
  e_t_data        type standard table  "Rückgabetabelle"
  e_end_of_data   type rs_bool         "letztes Datenpaket"
  e_t_msg         type rs_t_msg        "Meldungen an Frontend-Tool"

```

Ist der Funktionsbaustein so ausgelegt, dass er die Selektion auf Hierarchieknoten unterstützt, so muss er zwingend auch die Einschränkung von Merkmalen durch deren SID-Werte (anstelle der Merkmalsausprägungen) unterstützen.

Diese Unterstützung wird durch die Option *SID-Unterstützung* aktiviert. Wird diese Option verwendet, so kann die Analytical Engine auf die Umwandlung von SIDs in Merkmalswerte und umgekehrt verzichten, was entsprechende Performance-Vorteile mit sich bringt. Die Unterstützung von SIDs durch den Funktionsbaustein hat auch eine Veränderung der Schnittstelle des Funktionsbausteins zur Folge:

SID-Unterstützung

```

IMPORTING
  i_infoprov      type rsinfoprov      "InfoProvider"
  i_th_sfc        type rsdd_th_sfc      "benötigte Merkmale"
  i_th_sfk        type rsdd_th_sfk      "benötigte Kennzahlen"
  i_tsx_selldr    type rsdd_tsx_selldr  "Selektionsbedingungen (SID)"
  i_first_call    type rs_bool          "erster Aufruf des Bausteins"
  i_packagesize   type i                "Paketgröße"
EXPORTING
  e_t_data        type standard table   "Rückgabetablelle"
  e_end_of_data   type rs_bool          "letztes Datenpaket"
  e_t_msg         type rs_t_msg         "Meldungen an Frontend-Tool"

```

Der Funktionsbaustein für den VirtualProvider kann nicht nur im BW-System aufgerufen werden, sondern in jedem anderen System, das eine BAPI-Schnittstelle besitzt (also auch Nicht-ERP-Systeme).

RFC verpacken

Zu diesem Zweck muss ein entsprechendes logisches System festgelegt werden, in dem der Funktionsbaustein per RFC aufgerufen werden soll. Darüber hinaus werden die Parametertabellen zum Aufruf des Bausteins in ein BAPI-Format verpackt, was eine entsprechende Änderung der Schnittstelle nach sich zieht:

```

IMPORTING
  infocube        type rsinfoprov      "InfoProvider"
EXPORTING
  return          type bapiret2        "Returnparameter (OK=0)"
TABLES
  selection       type bapi6200s1      "Selektionsbedingungen"
  characteristics type bapi6200fd      "benötigte Merkmale"
  keyfigures      type bapi6200fd      "benötigte Kennzahlen"
  data            type bapi6100da      "generische Datenstruktur"
EXCEPTIONS
  communication_failure
  system_failure

```

Bei der Verwendung des RFC-Aufrufes werden ausschließlich Merkmalswerte ausgetauscht und keine SID-Werte. Die Option *SID-Unterstützung* kann nicht in Verbindung mit dem RFC verwendet werden.

27.6.2 VirtualProvider mit BAPI

Durch VirtualProvider mit BAPI³⁰ können Daten aus den Systemen von Drittanbietern bezogen werden. Das Prinzip ist dabei ähnlich dem VirtualProvider für Direktzugriff³¹, jedoch werden die erhaltenen Daten nicht mehr durch eine Transformation verarbeitet, sondern müssen durch das Quellsystem bereits in der Form geliefert werden, wie sie bei der Analyse benötigt werden. Die Datenanforderung wird auch nicht an die DataSources des Quellsystems gerichtet, sondern erfordert ein für diese Aufgabe vorbereitetes Third-Party-Quellsystem, das ein entsprechendes BAPI zum Austausch der Daten bereitstellt.

Das Anlegen eines VirtualProviders mit BAPI erfolgt analog zum Anlegen eines VirtualProviders für Direktzugriff, wobei jedoch der Typ *basierend auf einem BAPI* auszuwählen und die RFC-Destination des entsprechenden Third-Party-Quellsystems anzugeben ist³². Virtual-Provider mit BAPI werden in Spezialfällen eingesetzt, wenn Daten von einem Marktdatenanbieter (zum Beispiel AC Nielsen oder Dun & Bradstreet) bezogen werden sollen.

27.6.3 Virtuelle InfoProvider in BWA und HANA

Der Einsatz eines VirtualProviders ist in der Regel mit ganz besonderen Herausforderungen im Bereich des Performance-Tunings verbunden. Insbesondere wenn ein BWA oder HANA im Einsatz ist, werden unter Umständen zwar Lesezugriffe auf dem Datenbanksystem performant ausgeführt, der Datentransfer zum BW-Applikationsserver und die dortige Aufbereitung der Daten kann dennoch zu schlechten Antwortzeiten führen.

Die Antwortzeiten können durch BWA oder HANA nicht verbessert werden, solange die volle Funktionalität eines VirtualProviders beibehalten werden soll. Unter Umstände bietet jedoch die Indizierung der Ergebnisse als sogenannte *Snapshots*, also als Momentaufnahme eines Datenstands, eine Lösung. Inhaltlich setzt dies voraus, dass die Funktionsweise eines VirtualProviders durch einen Datenbestand ersetzt werden kann; dies bedeutet vor allem, dass sich Kennzahlwerte in jedem Navigationsschritt durch eine der angebotenen Aggregationen errechnen lassen.

30. Bis zur Version 3.x des BW als allgemeiner RemoteCube bezeichnet.

31. Siehe Abschnitt 22.1.

32. Siehe Abbildung 22-2 auf Seite 461.

Erfüllen VirtualProvider aus inhaltlicher Sicht alle Voraussetzungen, um als Datenbestand in einem Snapshot abgelegt zu werden, so kann die Funktionalität des VirtualProviders in der Regel ebenso in der Datenbewirtschaftung abgebildet werden. Überprüfen Sie vor der Indizierung derartiger VirtualProvider daher stets die Möglichkeit, stattdessen einen normalen InfoCube mit den entsprechenden Daten zu bewirtschaften.



Der Inhalt eines VirtualProviders kann mithilfe der Transaktion RSDDB in einem BWA-Index abgelegt werden (auch bei Verwendung von HANA). Das Modell dieses Index ist vergleichbar mit dem eines BWA-basierten InfoCubes, d.h., alle Merkmale des VirtualProviders sind zusammen mit den Kennzahlen flach in einer Faktentabelle indiziert und mit den Indizes für Stammdaten verbunden.

Anders als bei BWA-basierten InfoCubes wird der entsprechende BWA-Index jedoch nicht automatisch gesplittet³³. Stattdessen muss beim Anlegen des Index in der Transaktion RSDDB eingetragen werden, wie viele Datensätze indiziert werden. Die eingetragene Anzahl muss nicht genau sein und dient lediglich der Entscheidung, ob der Index gesplittet wird oder nicht.

Sobald ein VirtualProvider durch die Transaktion RSDDB indiziert wurde, werden Funktionsbaustein/BADI nur noch zur Neuindizierung und Aktualisierung des Index verwendet. Zugriffe durch die Analytical Engine hingegen lesen ausschließlich Daten aus dem BWA-Index.

Um die Indizierung von VirtualProvidern zu beschleunigen, kann der Vorgang durch die Implementierung von zwei Interfaces unterstützt werden:

- IF_RSDDB_BVIP_DELTA, um die Aktualisierung durch ein Delta-Verfahren zu ermöglichen,
- IF_RSDDB_BVIP_PARALLEL, um die initiale Indizierung des VirtualProviders zu parallelisieren.

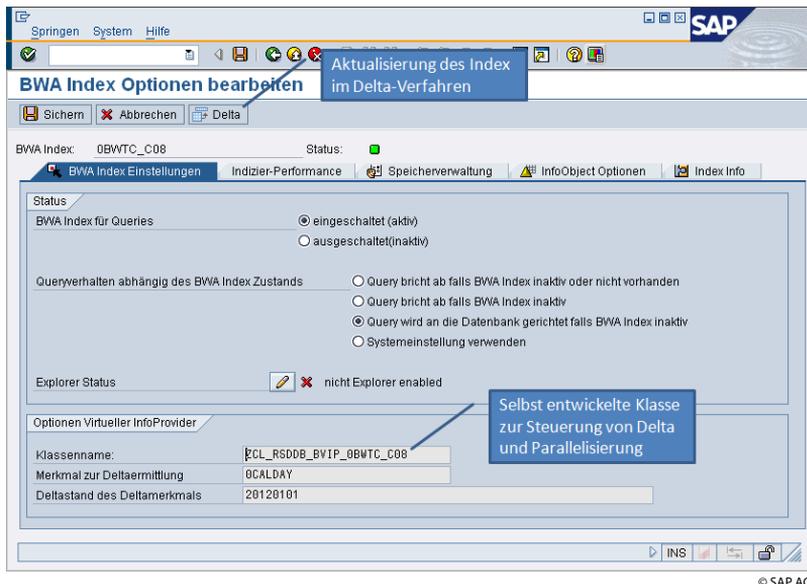
Für beide Interfaces existiert eine Beispielimplementierung in der Klasse CL_RSDDB_BVIP_SUPER. Diese kann für eine eigene Implementierung der Interfaces kopiert und beim Anlegen des Index in dessen Einstellungen angegeben werden (siehe Abb. 27–23).

Zur Aktualisierung eines Index für einen VirtualProvider muss dieser über ein Merkmal verfügen, das monoton aufsteigt und an dessen Inhalt neue Daten zu erkennen sind (Änderungen werden nicht unterstützt). Dieses Merkmal muss in den Einstellungen zum Index im Feld *Merkmal zur Deltaermittlung* angegeben werden.

*Aktualisierung im
Delta-Verfahren*

33. Siehe Abschnitt 12.3.

Abb. 27-23
VirtualProvider in
BWA und HANA



Im Interface IF_RSDDDB_BVIP_DELTA ist in diesem Fall die Methode GET_READPOINTER zu implementieren. Diese muss bei jeder Aktualisierung die Obergrenze der zu lesenden Werte für das deltabestimmende Merkmal im Parameter R_READPOINTER liefern³⁴. Der Wert der letzten erfolgreichen Indizierung kann in den Indexeinstellungen am Feld *Deltastand des Deltamerkmals* abgelesen werden.

Bei jeder Aktualisierung wird auf alle Werte selektiert, die zwischen dem *Deltastand des Deltamerkmals* (inkl.) und der neu ermittelten Obergrenze (exkl.) liegen. Wird zur Delta-Ermittlung ein Datum verwendet, so wird auf diese Weise sichergestellt, dass Daten nicht doppelt indiziert werden, wenn die Aktualisierung mehrmals am selben Tag aufgerufen wird³⁵.

Neuaufbau und Aktualisierung eines als Snapshot indizierten VirtualProviders kann in der Transaktion RSDDDB oder durch das Programm RSDDDB_BIAVIP_MAIN erfolgen.

34. Es wird exklusive der Obergrenze gelesen.

35. Wird ein Index bspw. am 01.01.2014 ein zweites Mal aktualisiert, so werden beim zweiten Mal alle Sätze gelesen, bei denen das deltabestimmende Datum die Bedingung $\geq 01.01.2014$ und $< 01.01.2014$ erfüllt – es werden also keine weiteren Datensätze indiziert.

Insbesondere wenn die Aktualisierung eines als Snapshot indizierten VirtualProviders nicht durch ein Delta-Verfahren möglich ist, muss die Performance der Neuindizierung optimiert werden. Dies kann erfolgen, indem die Daten des VirtualProviders nicht in einem Lesevorgang ermittelt, sondern mehrere Lesevorgänge parallel ausgeführt werden.

*Parallelisierung beim
Neuaufbau*

Wie viele Lesevorgänge parallel ausgeführt werden sollen und mit welcher Selektion jeder einzelne dieser Vorgänge Daten lesen soll, kann in der Methode `GET_RANGES` des Interface `IF_RSDDDB_BVIP_PARALLEL` bestimmt werden. Beim Rückgabeparameter `R_T_RANGE` handelt es sich um eine interne Tabelle mit Datensätzen der Struktur `RSDRI_S_RANGE`; für jeden Datensatz in dieser Tabelle wird ein Leseprozess ausgeführt. Die entsprechende Selektion ist in den einzelnen Datensätzen abzulegen. Das System überprüft dabei nicht, ob die Selektionen sich ggf. überlappen oder Wertebereiche auslassen.