

Core Data Services für ABAP

CDS-Views und Datenmodelle für SAP S/4HANA –
aktuell für Release 2023

» Hier geht's
direkt
zum Buch

DIE LESEPROBE

Kapitel 3

CDS-Assoziationen

In diesem Kapitel erfahren Sie, wie Sie Assoziationen in CDS-Modellen definieren. Außerdem lernen Sie, wie Sie Assoziationen beim Erstellen Ihrer CDS-View-Hierarchien sowie in Ihrem ABAP-Code einsetzen.

Mit *Assoziationen* können Sie gerichtete Beziehungen zwischen CDS-Modellen modellseitig erfassen. Assoziationen können Sie darüber hinaus auch zur Implementierung der internen Logik Ihrer CDS-Views nutzen. Außerdem können Sie Assoziationen exponieren, damit die Verwender Ihrer CDS-Views diese ebenfalls verwenden können. Diese mögliche externe Nutzung beinhaltet dabei sowohl die Wiederverwendung der in den Assoziationsdefinitionen enthaltenen Logik innerhalb aufeinander aufbauender CDS-Views eines CDS-View-Stacks als auch die Verwendung der Assoziationen innerhalb von SQL-Selektionsanfragen in ABAP-Anwendungen.

Darüber hinaus können exponierte Assoziationen zur Dokumentation von CDS-Modellen dienen. Assoziationen sind außerdem als Träger zahlreicher Metadaten wertvoll für die sie verwendenden bzw. interpretierenden Frameworks. Diese leiten zahlreiche Funktionen aus den zugehörigen Metadaten ab. Beispielsweise können assoziierte *Dimensions-CDS-Views* als Quelle von Anzeigeattributen in analytischen Queries genutzt werden. Weiterführende Informationen zu den Dimensions-Views erhalten Sie in Abschnitt 10.2.4, »Analytische Dimensions-Views«.

Konzeptionell werden Datenfelder und Assoziationen aus Sicht der CDS-Sprache als gleichrangig betrachtet und gemeinschaftlich als *CDS-Elemente* bezeichnet. Implementierungsseitig gibt es jedoch einige Unterschiede zwischen Datenfeldern und Assoziationen, die wir im Folgenden erörtern werden.

Wir konzentrieren uns zunächst auf die Definition (siehe Abschnitt 3.1, »Assoziationsdefinitionen«) und die Exponierung von Assoziationen (siehe den gleichnamigen Abschnitt 3.2). Anschließend besprechen wir besondere Formen von CDS-Assoziationen. Diese bilden Kompositionsbeziehungen (siehe Abschnitt 3.3, »Modellierung von Kompositionsbeziehungen«) und m:n-Beziehungen (siehe Abschnitt 3.4, »Modellierung von m:n-Beziehun-

Verwendungs-
möglichkeiten von
Assoziationen

CDS-Elemente

Aufbau dieses
Kapitels

gen«) ab. Assoziationen können analog zu Feldern projiziert werden. Dies erläutern wir in Abschnitt 3.5, »Projizierte Assoziationen«. Schließlich erfahren Sie, wie Sie CDS-Assoziationen in CDS-Views (siehe Abschnitt 3.6) und ABAP-Coding (Abschnitt 3.7) verwenden können.



Technische Abbildung der Assoziationen

Technisch werden Assoziationen auf Join-Verknüpfungen abgebildet. Diese Verknüpfungen kommen allerdings erst dann zum Tragen, wenn die Assoziationen tatsächlich verwendet werden. Dies ist z. B. dann der Fall, wenn mit ihrer Hilfe ein Pfadausdruck zu einem Feld gebildet wird. Die reine Definition oder Propagation einer Assoziation im CDS-View-Stack manifestiert sich ansonsten nicht in den aus den CDS-Views generierten Datenbank-Views.

3.1 Assoziationsdefinitionen

Die *Assoziationsdefinition* beschreibt die Verknüpfung der jeweiligen Definitionsquelle mit einem Assoziationsziel auf Ebene der korrelierten Datensätze. In Listing 3.1 werden beispielsweise Datensätze des CDS-Views `ZI_SalesOrder` mit den Datensätzen des Ziel-CDS-Views `ZI_SalesOrderItem` über die Assoziation `_Item` verknüpft.

```
define view entity ZI_SalesOrder
  as select from ...
  association [0..*] to ZI_SalesOrderItem as _Item
    on $projection.SalesOrder = _Item.SalesOrder
  ...
```

Listing 3.1 Assoziationsdefinition

Assoziationsnamen

Der Name einer Assoziation kann mit der Aliasfunktion `AS` definiert werden. Im zuvor skizzierten Beispiel lautet der Name `_Item`.



Namensraum der Assoziationen beachten

Die Namen der CDS-Elemente (Felder und Assoziationen) sowie die Parameternamen teilen sich denselben Namensraum. Das heißt, sie müssen eindeutig innerhalb eines CDS-Modells definiert werden. Sie sollten daher Namen von Assoziationen von Feldnamen und Parameternamen abgrenzen. Wir empfehlen Ihnen, zu diesem Zweck dem semantischen Namen der Assoziation einen Unterstrich als Präfix voranzustellen, wie bei `_Item` in unserem Beispiel.

Die Assoziationsdefinition enthält neben dem Namen des verknüpften Ziels auch die Beziehung zwischen den Datensätzen der Quell- und Zielentität. Diese relationalen Zusammenhänge werden mithilfe einer *On-Bedingung* erfasst.

**On-Bedingungen
der Assoziationen**

Im Beispiel aus Listing 3.1 wird dabei das Feld `SalesOrder` der Projektionsliste des Quell-CDS-Views `ZI_SalesOrder` über den Ausdruck `$projection.SalesOrder` angesprochen und durch den Pfadausdruck `_Item.SalesOrder` mit dem gleichnamigen Feld des Ziel-CDS-Views `ZI_SalesOrderItem` der Assoziation verknüpft.

Wird die Assoziation im Fall einer Navigation von der Definitionsquelle zum Ziel-CDS-View genutzt, findet eine der On-Bedingung entsprechende Filterung der Datensätze der Zielentität auf der Basis der Quelldatensätze statt.

Prinzipiell kann es zu einem Datensatz der Quelle einer Assoziation, abhängig von den festgelegten On-Bedingungen, unterschiedlich viele korrespondierende Datensätze des Assoziationsziels geben.

**Kardinalitäts-
angaben von
Assoziationen**

Umgekehrt kann es zu einem Zieldatensatz unterschiedlich viele Quelldatensätze geben. Die mögliche Anzahl an korrespondierenden Datensätzen wird über eine Kardinalitätsangabe dokumentiert.

Dabei gibt es zwei mögliche Arten der Erfassung:

- Eine Kardinalitätsangabe in eckigen Klammern [...] (siehe Listing 3.1) beschreibt, wie viele Datensätze des Assoziationsziels minimal und maximal mit einem Datensatz der Quelle verknüpft sein können.
- Eine Kardinalitätsangabe der Form [OF [EXACT] ONE/MANY] TO [EXACT] ONE/MANY erlaubt es Ihnen, nicht nur die Zielkardinalität, sondern auch die Quellkardinalität zu modellieren.

Kardinalität von Assoziationen angeben

Sie sollten die Kardinalität einer Assoziation immer explizit und so präzise wie möglich spezifizieren. Geben Sie dabei sowohl den Minimal- als auch den Maximalwert an. Dies erhöht die Lesbarkeit Ihres CDS-Modells und erlaubt es, dem SAP-HANA-Optimierer den effizientesten Ausführungsplan für Ihre Selektionsanfrage zu ermitteln.



Tabelle 3.1 zeigt Ihnen einige Beispiele der in CDS-Modellen vorzufindenden Ausprägungen der Kardinalitätsangabe.

Kardinalitäts- angabe	Anzahl der korrelierten Datensätze			
	Assoziationsquelle		Assoziationsziel	
	Minimal	Maximal	Minimal	Maximal
[1]	0	unbegrenzt	0	1
[0..1]	0	unbegrenzt	0	1
[1..1]	0	unbegrenzt	1	1
[0..*]	0	unbegrenzt	0	unbegrenzt
[1..*]	0	unbegrenzt	1	unbegrenzt
TO ONE	0	unbegrenzt	0	1
TO EXACT ONE	0	unbegrenzt	1	1
TO MANY	0	unbegrenzt	0	unbegrenzt
OF ONE TO ONE	0	1	0	1
OF EXACT ONE TO EXACT ONE	1	1	1	1
OF MANY TO MANY	0	unbegrenzt	0	unbegrenzt
OF MANY TO ONE	0	unbegrenzt	0	1
OF MANY TO EXACT ONE	0	unbegrenzt	1	1
keine (Default-Logik)	0	unbegrenzt	0	1

Tabelle 3.1 Kardinalitätsangaben in Assoziationsdefinitionen



Kardinalitätsangaben überprüfen

Sie sollten die Kardinalitätsangaben Ihrer Assoziationsdefinitionen kritisch prüfen. Fehlerhafte Kardinalitätsangaben können zu Folgefehlern in der Datenmodellierung führen. Ferner können daraus sowohl Duplikate als auch fehlende Datensätze sowie fehlerhafte Daten in den Selektionsergebnissen resultieren. Nicht zuletzt drohen bei fehlerhaften Kardinalitätsangaben Performance-Einbußen.

Speziell bei der Verwendung der Exact-One-Semantik sollten Sie prüfen, ob diese auch bei Anwendung außerhalb des CDS-Views, der die Assoziation definiert, gültig ist.

Generell unterstützen verschiedene Arten von CDS-Modellen die Definition von Assoziationen. Beispielsweise können Assoziationen in CDS-Views und deren Erweiterungen definiert werden. Technisch zulässige Assoziationsziele sind unter anderem:

Assoziationsquellen
und -ziele

- CDS-Views
- CDS-Tabellenfunktionen
- Abstrakte CDS-Entitäten
- Datenbanktabellen

Assoziationsziele wählen

Sie sollten Assoziationen vornehmlich zwischen gleichartigen CDS-Modellen definieren, um funktionale Einschränkungen zu vermeiden. Beispielsweise können Datenbanktabellen in analytischen Anwendungen nicht als Ziel von Fremdschlüsselassoziationen genutzt werden.



3.2 Exponierung von Assoziationen

Mit der Definition einer Assoziation steht diese für die interne Implementierung eines CDS-Views bereit. Um die Assoziation auch außenstehenden Verwendern zugänglich zu machen, muss diese analog zu den Feldern in die Projektionsliste des CDS-Views aufgenommen werden. Ohne diese Exponierung handelt es sich bei der Assoziationsdefinition lediglich um ein internes Implementierungsdetail, dessen Funktion Sie auch über die Definition von entsprechenden Join-Verknüpfungen hätten realisieren können.

In Listing 3.2 wird die lokal definierte Assoziation `_Item` in die Projektionsliste übernommen. Sie ist damit Bestandteil der externen Signatur des CDS-View-Modells und kann von dessen Verwendern angesprochen werden.

Beispiel: CDS-View
mit exponierter
Assoziation

```
define view entity ZI_SalesOrder
  as select from ...
  association [0..*] to ZI_SalesOrderItem as _Item
    on $projection.SalesOrder = _Item.SalesOrder
{
  key SalesOrder,
    _Item,
    ...
}
```

Listing 3.2 CDS-View mit exponierter Assoziation



Nicht genutzte Assoziationen entfernen

Definierte Assoziationen, die im CDS-View-Modell weder zur Implementierung der View-Logik verwendet noch in dessen Signatur exponiert werden, bieten keine Funktion. Als Implementierungsbestandteil werden sie jedoch im Rahmen der CDS-Aktivierung geprüft. Sie sollten derartige Assoziationen löschen, um Aktivierungsprobleme zu vermeiden.

Beachten Sie, dass Assoziationen zu *Extension-Include-Views* im CDS-View nur definiert, aber nicht exponiert werden (siehe Kapitel 15, »Erweiterungen von CDS-Views und weiteren Entitäten«). Sie werden von der Extensibility-Infrastruktur zur Bildung von Pfadausdrücken genutzt.

3.3 Modellierung von Kompositionsbeziehungen

Kompositionsbeziehungen repräsentieren eine Spezialisierung der Assoziationen. Sie erfassen eine existenzielle Abhängigkeit eines Kompositionskindes von dessen Kompositionsvater. Beispielsweise gehört eine Auftragsposition (Kompositionskind) immer zu einem Auftragskopf (Kompositionsvater).

Kompositionen können mit bestimmten CDS-Syntaxelementen definiert werden. Wir erläutern die entsprechenden CDS-Syntaxelemente in den Beispiel-CDS-Views in Listing 3.3 bis Listing 3.5.

Beispiel:
Wurzel-CDS-View

Listing 3.3 zeigt den Wurzel-CDS-View `Z_CompositionRootView` einer Kompositionshierarchie.

```
define root view entity Z_CompositionRootView
  as select distinct from t000
  composition [0..*] of Z_CompositionChildView as _ChildView
{
  key abap.char'R' as RootKeyField,
  _ChildView
}
```

Listing 3.3 Wurzel-CDS-View mit Kompositionsbeziehung

Dieser Wurzel-CDS-View wird durch das Schlüsselwort `ROOT` als solcher ausgezeichnet. Kompositionsbeziehungen werden mit `COMPOSITION ... OF` definiert und spannen ausgehend vom Wurzel-CDS-View die weitere Kompositionshierarchie auf.

In unserem Fall wird der CDS-View `Z_CompositionChildView` aus Listing 3.4 über die Kompositionsbeziehung `_ChildView` als Kind-CDS-View in diese Hierarchie aufgenommen.

Beispiel: Kind-CDS-View aus einer Kompositionshierarchie

```
define view entity Z_CompositionChildView
  as select distinct from t000
  association to parent Z_CompositionRootView as _RootView
    on $projection.RootKeyField = _RootView.RootKeyField
  composition [0..*] of Z_CompositionGrandchildView
    as _GrandchildView
{
  key abap.char'R' as RootKeyField,
  key abap.char'C' as ChildKeyField,
  _RootView,
  _GrandchildView
}
```

Listing 3.4 CDS-View, der das Kompositionskind definiert

Das Kompositionskind muss dabei eine korrespondierende *To-Parent-Assoziation* zu seinem Kompositionsvater definieren und exponieren. Aufgrund der existenziellen Abhängigkeit entspricht die minimale und maximale Zielkardinalität dieser Assoziation immer dem Wert »1«. Die Quellkardinalität der To-Parent-Assoziation folgt unmittelbar aus der gegenläufigen Kompositionsbeziehung. Die explizite Angabe der Kardinalität in der Definition der To-Parent-Assoziation entfällt daher.

Die On-Bedingung der To-Parent-Assoziation muss alle Schlüsselfelder der assoziierten Vaterentität binden. Sie spezifiziert sowohl die Beziehung der Datensätze des Kind-CDS-Views zu dessen Vater-CDS-View als auch die Beziehung der Datensätze des Vater-CDS-Views zu dessen Kind-CDS-View. Das heißt, die On-Bedingung der Assoziation `_RootView` in Listing 3.4 legt gleichsam die On-Bedingung der Kompositionsbeziehung `_ChildView` in Listing 3.3 fest. Letztere wird daher nicht explizit spezifiziert.

Kompositionsbeziehungen in abstrakten Entitäten

In abstrakten Entitäten können Sie auf die Definition der On-Bedingungen von Kompositionsbeziehungen gänzlich verzichten.



Der in Listing 3.4 skizzierte Kind-CDS-View `Z_CompositionChildView` selbst definiert über die Kompositionsbeziehung `_GrandchildView` den Enkel-CDS-View `Z_CompositionGrandchildView` aus Listing 3.5 als weitere Hierarchiestufe.

Beispiel: Enkel-CDS-View aus einer Kompositionshierarchie


```

define view entity Z_CompositionGrandchildView
  as select distinct from t000
  association [1..1] to Z_CompositionRootView as _RootView
    on $projection.RootKeyField = _RootView.RootKeyField
  association to parent Z_CompositionChildView as _ChildView
    on $projection.RootKeyField = _ChildView.RootKeyField
    and $projection.ChildKeyField = _ChildView.ChildKeyField
{
  key abap.char'R' as RootKeyField,
  key abap.char'C' as ChildKeyField,
  key abap.char'G' as GrandchildKeyField,
  _RootView,
  _ChildView
}

```

Listing 3.5 CDS-View, der den Kompositionsenkel definiert

In unserem Fall definiert der Enkel-CDS-View `Z_CompositionGrandchildView` die zusätzliche Assoziation `_RootView`, die eine direkte Verbindung zur Wurzelentität `Z_CompositionRootView` der Hierarchie realisiert. Diese Assoziation wird technisch nicht eingefordert. Sie finden derartige Assoziationen jedoch in vielen CDS-Modellen, da sie z. B. den Zugriff auf bzw. das Einbinden von Feldern der Wurzelentität für die Verwender Ihres Views sowie die Definition von Zugriffskontrollen (siehe Kapitel 5, »CDS-Zugriffskontrollen«) erleichtert.



Technische Einschränkungen beachten

Wenn Sie Kompositionsbeziehungen mit den entsprechenden Syntaxelementen modellieren, resultieren daraus einige Einschränkungen im Vergleich zu den Nutzungsmöglichkeiten von gewöhnlichen Assoziationen. Beispielsweise dürfen Sie die Kompositionsbeziehungen innerhalb des sie definierenden CDS-Views nicht zur Definition von Pfadausdrücken verwenden. Weiterhin unterliegen die Kompositionsbeziehungen zusätzlichen Konsistenzbedingungen: Kompositionsväter müssen auf ihre Kinder verweisen und umgekehrt, der Wurzel-View muss mit dem Schlüsselwort `ROOT` ausgezeichnet werden usw. Diese Konsistenzbedingungen müssen erfüllt sein, damit Sie die CDS-Modelle erfolgreich aktivieren können.

3.4 Modellierung von m:n-Beziehungen

Falls ein Datensatz einer Entität Beziehungen zu vielen Datensätzen einer anderen Entität aufweist und umgekehrt, stehen die beiden Entitäten in einer *m:n-Beziehung* zueinander. Wenn Sie derartige m:n-Beziehungen in Ihren CDS-Views abbilden wollen, müssen Sie in der Regel einen eigenständigen CDS-View modellieren, der die m:n-Beziehung erfasst. Dank dieser Vorgehensweise müssen Sie nicht den Quell- oder aber den Ziel-CDS-View der Relation mit der m:n-Beziehungsinformation per Join anreichern. Würden Sie stattdessen die m:n-Beziehung direkt im Quell- bzw. Ziel-CDS-View definieren, müssten Sie unerwünschte Kardinalitätssprünge in Kauf nehmen.

Separate CDS-Views zur Definition von m:n-Beziehungen

Betrachten Sie dazu das folgende Beispiel. Es umfasst die drei CDS-Views *Z_MToNRelationViewA* aus Listing 3.6, *Z_MToNRelationViewB* aus Listing 3.7 und *Z_MToNRelationViewC* aus Listing 3.8. Abbildung 3.1 illustriert das Beziehungsgeflecht dieser CDS-Views.

Beispiel: CDS-Views mit m:n-Beziehung

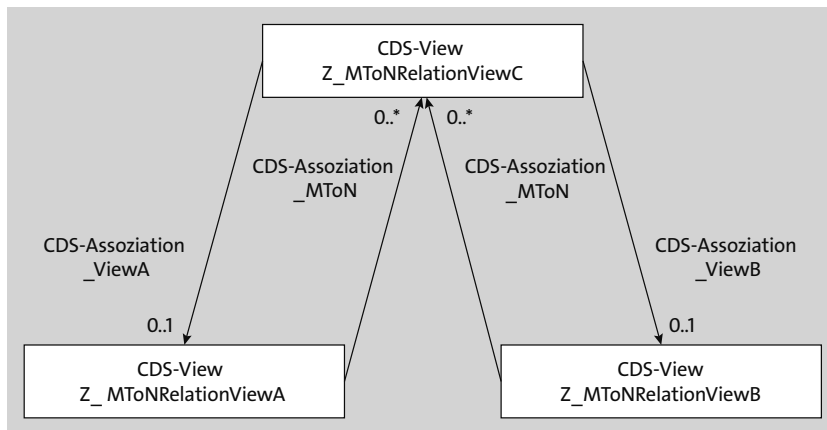


Abbildung 3.1 Assoziationen der CDS-Views, die über eine m:n-Beziehung miteinander in Verbindung stehen

Der CDS-View *Z_MToNRelationViewA* aus Listing 3.6 (im Folgenden kurz *View A* genannt) besitzt ein einzelnes Schlüsselfeld *KeyFieldA*. Seine Assoziation *_MToN* nutzt dieses Schlüsselfeld und verknüpft den View A mit dem CDS-View *Z_MToNRelationViewC* aus Listing 3.8.

Beispiel: View A

```
define view entity Z_MToNRelationViewA
  as select distinct from t000
  association [0..*] to Z_MToNRelationViewC as _MToN
  on $projection.KeyFieldA = _MToN.KeyFieldA
{
```

```

    key abap.int4'1' as KeyFieldA,
      _MToN
  }
union select distinct from t000
  association [0..*] to Z_MToNRelationViewC as _MToN
  on $projection.KeyFieldA = _MToN.KeyFieldA
{
  key abap.int4'2' as KeyFieldA,
    _MToN
}

```

Listing 3.6 CDS-View »Z_MToNRelationViewA«

Der CDS-View Z_MToNRelationViewA (View A) liefert zwei Datensätze entsprechend Tabelle 3.2, die einen Überblick über die von den drei CDS-Views dieses Beispiels bereitgestellten Datensätze vermittelt.

CDS-View	KeyFieldA	KeyFieldB
Z_MToNRelationViewA	1	–
	2	–
Z_MToNRelationViewB	–	3
	–	4
Z_MToNRelationViewC	1	3
	1	4
	2	4

Tabelle 3.2 Datensätze der CDS-Views, die über eine m:n-Beziehung miteinander in Verbindung stehen

Beispiel: View B

Analog ist der CDS-View Z_MToNRelationViewB aus Listing 3.7 (im Folgenden kurz *View B* genannt) mit dem Schlüssel KeyFieldB über eine gleichnamige Assoziation _MToN mit dem CDS-View Z_MToNRelationViewC aus Listing 3.8 verknüpft. Auch der CDS-View Z_MToNRelationViewB liefert entsprechend Tabelle 3.2 zwei Datensätze.

```

define view entity Z_MToNRelationViewB
  as select distinct from t000
  association [0..*] to Z_MToNRelationViewC as _MToN
  on $projection.KeyFieldB = _MToN.KeyFieldB
{

```

```

key abap.int4'3' as KeyFieldB,
    _MToN
}
union select distinct from t000
    association [0..*] to Z_MToNRelationViewC as _MToN
        on $projection.KeyFieldB = _MToN.KeyFieldB
{
    key abap.int4'4' as KeyFieldB,
        _MToN
}

```

Listing 3.7 CDS-View »Z_MToNRelationViewC«

Der assoziierte CDS-View Z_MToNRelationViewC aus Listing 3.8 besitzt einen kombinierten Schlüssel KeyFieldA und KeyFieldB. Er bildet eine m:n-Beziehung zwischen den Views A und B aus Listing 3.6 und Listing 3.7 ab. Er definiert dazu die Assoziationen _ViewA und _ViewB zu diesen beiden CDS-Views. Diese gestatten es den Verwendern des Views Z_MToNRelationViewC, die Datensätze des Views A mit den zugehörigen Datensätzen des Views B zu korrelieren und umgekehrt.

Beispiel: CDS-View zur Definition der m:n-Beziehung

```

define view entity Z_MToNRelationViewC
    as select distinct from t000
        association [0..1] to Z_MToNRelationViewA as _ViewA
            on $projection.KeyFieldA = _ViewA.KeyFieldA
        association [0..1] to Z_MToNRelationViewB as _ViewB
            on $projection.KeyFieldB = _ViewB.KeyFieldB
{
    key abap.int4'1' as KeyFieldA,
    key abap.int4'3' as KeyFieldB,
        _ViewA,
        _ViewB
}
union select distinct from t000
    association [0..1] to Z_MToNRelationViewA as _ViewA
        on $projection.KeyFieldA = _ViewA.KeyFieldA
    association [0..1] to Z_MToNRelationViewB as _ViewB
        on $projection.KeyFieldB = _ViewB.KeyFieldB
{
    key abap.int4'1' as KeyFieldA,
    key abap.int4'4' as KeyFieldB,
        _ViewA,
        _ViewB
}

```

```

}
union select distinct from t000
  association [0..1] to Z_MToNRelationViewA as _ViewA
    on $projection.KeyFieldA = _ViewA.KeyFieldA
  association [0..1] to Z_MToNRelationViewB as _ViewB
    on $projection.KeyFieldB = _ViewB.KeyFieldB
{
  key abap.int4'2' as KeyFieldA,
  key abap.int4'4' as KeyFieldB,
  _ViewA,
  _ViewB
}

```

Listing 3.8 CDS-View »Z_MToNRelationViewC«

**Analyse der
Beispiel-Views**

Der CDS-View `Z_MToNRelationViewC` liefert insgesamt drei Datensätze entsprechend Tabelle 3.2. Diese setzen die jeweils vorliegenden Datensätze der Views A und B miteinander in Verbindung. Wollten Sie diese Information in einen dieser beiden Views A oder B integrieren, müssten Sie dessen bestehende Datensätze entsprechend ausmultiplizieren.

Ein Verwender dieses CDS-View-Geflechts kann nun (je nach Quell-View A oder B) Pfadausdrücke der Form `_MToN._ViewB` oder `_MToN._ViewA` nutzen, um die miteinander verknüpften Datenbestände der beiden Views A und B auszuwerten.

3.5 Projizierte Assoziationen

**Assoziationen
projizieren**

Exponierte Assoziationen eines CDS-Views können prinzipiell genauso wie die Felder des Views in die Implementierung eines darauf aufbauenden CDS-Views einfließen. Beispielsweise können Sie die exponierten Assoziationen eines zugrunde liegenden Basis-CDS-Views in die Projektionsliste Ihres eigenen CDS-Views aufnehmen und damit über diesen CDS-View ebenfalls exponieren. Falls erforderlich, können Sie dabei die projizierten Assoziationen auch mit Aliasnamen versehen.

**Beispiel: CDS-View
mit projizierter
Assoziation**

Listing 3.9 illustriert ein Beispiel für einen CDS-View mit projizierter Assoziation.

```

define view entity ZC_SalesOrder
  as select from ZI_SalesOrder
{

```

```

key ZI_SalesOrder.SalesOrder,
  ZI_SalesOrder._Item as _SalesOrderItem
}

```

Listing 3.9 CDS-View mit projizierter Assoziation

Darin übernimmt der CDS-View ZC_SalesOrder durch Projektionsbildung die Assoziation _Item des Basis-CDS-Views ZI_SalesOrder aus Listing 3.2. Die Assoziation wird mit dem Aliasnamen _SalesOrderItem exponiert. Neben der Assoziation _Item muss dabei auch das darin eingehende Feld Sales Order in die Projektionsliste des CDS-Views ZC_SalesOrder aufgenommen werden.

**Analyse des
Beispiel-Views**

Abhängigkeiten projizierter Assoziationen beachten

Wenn Sie eine Assoziation durch Projektionsbildung in Ihren CDS-View übernehmen und exponieren, müssen Sie sicherstellen, dass auch alle Felder des Basis-Views, die in der On-Bedingung seiner Assoziationsdefinition verwendet werden, in die Projektionsliste Ihres CDS-Views mit aufgenommen werden. Sie dürfen diese Basisfelder bei Bedarf über Aliasnamen umbenennen.



3.6 Verwendung von Assoziationen in CDS-Views

Mithilfe der Assoziationen lassen sich Pfadausdrücke definieren. Deren Segmente werden durch Punkte voneinander abgegrenzt. Das letzte Segment dieser Pfade bildet entweder ein Feld oder eine Assoziation des Ziels der vorangestellten Assoziation. Sie erfahren in Abschnitt 3.6.1, »Pfadausdrücke definieren«, mehr über Pfadausdrücke.

Pfadausdrücke

Abhängigkeiten der Pfadausdrücke beachten

Wenn Sie mehrere Assoziationen in Pfadausdrücken zu Ketten verknüpfen, müssen Sie auch in diesem Fall alle Quellfelder, die in die On-Bedingungen der einzelnen Assoziationsdefinitionen eingehen, in Ihren CDS-View aufnehmen.



Aus technischer Sicht führen Pfadausdrücke zu impliziten Join-Verknüpfungen. Diese werden im Abschnitt 3.6.2, »Implizite Join-Verknüpfungen«, beschrieben. Diese impliziten Join-Verknüpfungen können die Kardinali-

tät des Selektionsergebnisses beeinflussen. In Abschnitt 3.6.3, »Kardinalitätsänderungen durch Pfadausdrücke«, erfahren Sie Näheres zu dieser Thematik. Wenn Sie Assoziationen auf Basis lokal berechneter Felder definieren, können Sie diese nicht dazu nutzen, Pfadausdrücke im selben CDS-View zu formulieren. Entsprechende Einschränkungen behandeln wir in Abschnitt 3.6.4, »Berechnete Felder in Assoziationsdefinitionen«.

3.6.1 Pfadausdrücke definieren

Betrachten Sie exemplarisch die CDS-Views aus Listing 3.10 bis Listing 3.13, die ein Geflecht miteinander assoziierter CDS-Views repräsentieren sowie den darauf aufbauenden CDS-View aus Listing 3.14, der Pfadausdrücke auf Basis dieses Assoziationsgeflechts definiert.

Beispiel: CDS-View der Kundenauftrags-einteilung

Der CDS-View `ZI_SalesOrderScheduleLine` aus Listing 3.10 beschreibt die Einteilung des Kundenauftrags. Er verfügt über die Assoziation `_SalesOrderItem` zum CDS-View `ZI_SalesOrderItem` der Kundenauftragsposition.

```
define view entity ZI_SalesOrderScheduleLine
  as select from zsalesordersline
  association [1..1] to ZI_SalesOrderItem as _SalesOrderItem
    on $projection.SalesOrder = _SalesOrderItem.SalesOrder
    and $projection.SalesOrderItem = _SalesOrderItem.SalesOrderItem
  {
    key salesorder as SalesOrder,
    key salesorderitem as SalesOrderItem,
    key salesorderscheduleline as SalesOrderScheduleLine,
    _SalesOrderItem
  }
}
```

Listing 3.10 CDS-View der Einteilung des Kundenauftrags

Beispiel: CDS-View der Kundenauftragsposition

Der CDS-View `ZI_SalesOrderItem` aus Listing 3.11 beschreibt die Kundenauftragsposition. Er verfügt über eine Assoziation `_Product` zum CDS-View `ZI_Product` des Produkts.

```
define view entity ZI_SalesOrderItem
  as select from zsalesorderitem
  association [0..1] to ZI_Product as _Product
    on $projection.Product = _Product.Product
  {
    key salesorder as SalesOrder,
    key salesorderitem as SalesOrderItem,
```

```

        product      as Product,
        _Product
    }

```

Listing 3.11 CDS-View der Kundenauftragsposition

Der CDS-View `ZI_Product` aus Listing 3.12 verfügt über eine Assoziation `_Text` zum CDS-View `ZI_ProductText` des Produkttexts.

Beispiel: CDS-View des Produkts

```

define view entity ZI_Product
  as select from zproduct
  association [0..*] to ZI_ProductText as _Text
  on $projection.Product = _Text.Product
{
  key product as Product,
  _Text
}

```

Listing 3.12 CDS-View des Produkts

Listing 3.13 beinhaltet die Definition des CDS-Views `ZI_ProductText`, der den Produkttext beschreibt.

Beispiel: CDS-View des Produkttexts

```

define view entity ZI_ProductText
  as select from zproducttext
{
  key language      as Language,
  key product       as Product,
  product_name      as ProductName
}

```

Listing 3.13 CDS-View des Produkttexts

Auf Grundlage dieser vier CDS-Views wird nun in Listing 3.14 der CDS-View `Z_ViewWithPathExpressions` definiert.

Beispiel: CDS-View mit Pfadausdrücken

```

define view entity Z_ViewWithPathExpressions
  as select from ZI_SalesOrderScheduleLine
{
  key SalesOrder,
  key SalesOrderItem,
  key SalesOrderScheduleLine,
  _SalesOrderItem,
  _SalesOrderItem.Product as SalesOrderItemProduct,
  _SalesOrderItem._Product,

```



```
        _SalesOrderItem._Product.Product,  
        _SalesOrderItem._Product._Text  
    }  
}
```

Listing 3.14 CDS-View der Kundenauftragseinteilung mit Pfadausdrücken

Analyse des Beispiels

Der CDS-View `Z_ViewWithPathExpressions` übernimmt die Assoziation `_SalesOrderItem` des CDS-Views `ZI_SalesOrderScheduleLine` aus Listing 3.10 in seine Projektionsliste. Dies erfordert gleichzeitig auch die Übernahme der Felder `SalesOrder` und `SalesOrderItem`, da diese in der On-Bedingung der Assoziationsdefinition des Basis-CDS-Views verwendet werden.

Über den Pfadausdruck `_SalesOrderItem._Product` wird eine Beziehung zum CDS-View `ZI_Product` aus Listing 3.12 hergestellt. Der CDS-View `Z_ViewWithPathExpressions` selbst besitzt damit auch eine exponierte Assoziation `_Product` zum Produktstamm. Damit diese Assoziation exponiert werden kann, muss auch das Feld `Product` aus dem CDS-View `ZI_SalesOrderItem` aus Listing 3.11 in die Projektionsliste übernommen werden, da dieses Feld in der On-Bedingung der zugrunde liegenden Assoziationsdefinition verwendet wird. Die Einbindung des Felds erfolgt über den Pfadausdruck `_SalesOrderItem.Product`. Um Namenskollisionen zu vermeiden, wird das damit aufgenommene Feld `Product` mit dem Aliasnamen `SalesOrderItemProduct` versehen.

Genauso wird nun die Assoziation zum Text-CDS-View des Produkts `ZI_ProductText` aus Listing 3.13 per Pfadausdruck `_SalesOrderItem._Product._Text` übernommen. Um die Assoziation `_Text` exponieren zu können, muss die Projektionsliste um das Feld `Product` des Views `ZI_Product` erweitert werden. Dies geschieht über den Pfadausdruck `_SalesOrderItem._Product.Product`.

3.6.2 Implizite Join-Verknüpfungen

Entstehung impliziter Join-Verknüpfungen

Bei der Aufnahme von Feldern über Pfadausdrücke werden implizit die in den Assoziationsdefinitionen hinterlegten On-Bedingungen in Join-Verknüpfungen überführt. Entsprechend resultieren aus den beiden Pfadausdrücken `_SalesOrderItem.Product` und `_SalesOrderItem._Product.Product` in Listing 3.14 zwei effektive Join-Verknüpfungen.



Entstehung impliziter Join-Verknüpfungen

Die Definition und Exponierung von Assoziationen allein führt nicht zu zusätzlichen Join-Verknüpfungen. Diese werden erst durch die Verwendung der Assoziationen in Pfadausdrücken erzeugt.

Listing 3.15 veranschaulicht dies anhand der Create-Anweisung zum Datenbank-View, der aus dem CDS-View aus Listing 3.14 generiert wird.

**Datenbank-View
mit implizit
generierten Joins**

```
CREATE OR REPLACE VIEW "Z_VIEWWITHPATHEXPRESSIONS" AS SELECT
  "ZI_SALESORDERSCHEDULELINE"."MANDT" AS "MANDT",
  "ZI_SALESORDERSCHEDULELINE"."SALESORDER",
  "ZI_SALESORDERSCHEDULELINE"."SALESORDERITEM",
  "ZI_SALESORDERSCHEDULELINE"."SALESORDERSCHEDULELINE",
  "=A0"."PRODUCT" AS "SALESORDERITEMPRODUCT",
  "=A1"."PRODUCT"
FROM (
  "ZI_SALESORDERSCHEDULELINE" "ZI_SALESORDERSCHEDULELINE"
  LEFT OUTER MANY TO ONE JOIN "ZI_SALESORDERITEM" "=A0" ON (
    "ZI_SALESORDERSCHEDULELINE"."SALESORDER" = "=A0"."SALESORDER" AND
    "ZI_SALESORDERSCHEDULELINE"."SALESORDERITEM" =
      "=A0"."SALESORDERITEM" AND
    "ZI_SALESORDERSCHEDULELINE"."MANDT" = "=A0"."MANDT"
  )
) LEFT OUTER MANY TO ONE JOIN "ZI_PRODUCT" "=A1" ON (
  "ZI_SALESORDERSCHEDULELINE"."MANDT" = "=A1"."MANDT" AND
  "=A0"."PRODUCT" = "=A1"."PRODUCT"
)
WHERE "ZI_SALESORDERSCHEDULELINE"."MANDT" = SESSION_CONTEXT(
  'CDS_CLIENT'
)
```

Listing 3.15 Create-Anweisung zur Anlage des Datenbank-Views zum CDS-View aus Listing 3.14

Auf Datenbankebene treten die in den Pfadausdrücken verwendeten Assoziationen `_SalesOrderItem` und `_Product` des CDS-Views `Z_ViewWithPathExpressions` aus Listing 3.14 nicht in Erscheinung. Vielmehr werden diese im generierten Datenbank-View `Z_VIEWWITHPATHEXPRESSIONS` durch Join-Verknüpfungen ersetzt. Diese übernehmen neben den On-Bedingungen auch die Kardinalitätsangaben der Assoziationen. Der Zusatz `TO ONE` signalisiert der Datenbank, dass der Join die Kardinalität des Ergebnisses der Daten Selektion nicht beeinflusst. Damit muss dieser nicht zwangsläufig ausgeführt werden, sofern die Felder `SalesOrderItemProduct` und `Product` im Rahmen der Datenselektion nicht angefordert bzw. benötigt werden. Die impliziten Join-Verknüpfungen wirken sich dennoch negativ auf die statische View-Komplexität aus und können die Optimierung der Prozessierung der Selektion auf der Datenbank behindern.

**Analyse des
Datenbank-Views**

Beispiel: CDS-View mit expliziten Join-Verknüpfungen

Alternativ zur Modellierung des CDS-Views `Z_ViewWithPathExpressions` aus Listing 3.14 mit Pfadausdrücken kann die gleiche Funktion auch ohne Pfadausdrücke allein durch die Nutzung von expliziten Join-Verknüpfungen realisiert werden. Dies zeigt Listing 3.16.

```
define view entity Z_ViewWithoutPathExpressions
  as select from          ZI_SalesOrderScheduleLine as SL
    left outer many to one join ZI_SalesOrderItem as ITEM
      on ITEM.SalesOrder   = SL.SalesOrder
      and ITEM.SalesOrderItem = SL.SalesOrderItem
    left outer many to one join ZI_Product as PROD
      on PROD.Product = ITEM.Product
{
  key SL.SalesOrder,
  key SL.SalesOrderItem,
  key SL.SalesOrderScheduleLine,
    SL._SalesOrderItem,
    ITEM.Product as SalesOrderItemProduct,
    ITEM._Product,
    PROD.Product,
    PROD._Text
}
```

Listing 3.16 CDS-View der Kundenauftragseinteilung mit expliziten Join-Verknüpfungen



Denormalisierungen der CDS-Modelle vermeiden

Sie sollten auf (potenziell implizite) Join-Verknüpfungen der Datenquellen so weit wie möglich verzichten, um die statische Komplexität der CDS-View-Modelle so gering wie möglich zu halten. Dies gilt insbesondere für Ihre zentralen, mehrfach verwendeten CDS-Views.

Anstatt die CDS-Modelle explizit oder implizit zu denormalisieren, sollten Sie versuchen, die entsprechenden Relationen der Datenquellen über Assoziationen zu erfassen und diese dem Verwender (eventuell verteilt über mehrere assoziierte CDS-Views) zur Verfügung zu stellen. Der Verwender kann dann zweckgerichtet die exponierten Assoziationen verfolgen und sich so das gesamte Netzwerk der miteinander in Beziehung stehenden Modelle selbst erschließen und dabei nur die tatsächlich notwendigen Join-Verknüpfungen realisieren.

Standardmäßig führen die in einem Pfadausdruck zu einem Feld verwendete Assoziationen zu einem Left Outer Join. Wird eine Assoziation oder ein Pfadausdruck indes innerhalb der From-Anweisung eines CDS-Views genutzt, wird diese Assoziation bzw. dieser Pfadausdruck als Inner Join interpretiert.

Typen implizit generierter Join-Verknüpfungen

Listing 3.17 zeigt dies anhand eines Beispiels. Der darin definierte CDS-View `Z_ViewWithPathExprDataSources` verwendet, ausgehend vom CDS-View `ZI_SalesOrderItem`, die Assoziation `_SalesOrder`, um seine Datenquellen zu definieren und Elemente des Ziel-CDS-Views `ZI_SalesOrder` zu projizieren.

Beispiel: CDS-View mit assoziierter Datenquelle

```
define view entity Z_ViewWithPathExprDataSources
  as select from ZI_SalesOrderItem._SalesOrder
{
  key SalesOrder,
    _Item as _SalesOrderItem
}
```

Listing 3.17 CDS-View mit Pfadausdruck als Datenquelle

In diesem Fall dient als Datenquelle eine Inner-Join-Verknüpfung des Views `ZI_SalesOrderItem` mit dem View `ZI_SalesOrder`, entsprechend Listing 3.18.

Analyse des Beispiels

```
CREATE OR REPLACE VIEW "Z_VIEWWITHPATHEXPRDATASOURCES" AS SELECT
  "=AO"."MANDT" AS "MANDT",
  "ZI_SALESORDER"."SALESORDER"
FROM "ZI_SALESORDERITEM" "=AO" INNER JOIN "ZI_SALESORDER"
  "ZI_SALESORDER" ON (
  "=AO"."MANDT" = "ZI_SALESORDER"."MANDT" AND
  "=AO"."SALESORDER" = "ZI_SALESORDER"."SALESORDER"
)
WHERE "=AO"."MANDT" = SESSION_CONTEXT(
  'CDS_CLIENT'
)
```

Listing 3.18 Create-Anweisung zur Anlage des Datenbank-Views zum CDS-View aus Listing 3.17

3.6.3 Kardinalitätsänderungen durch Pfadausdrücke

Pfadausdrücke können die Anzahl der selektierten Datensätze und damit die Kardinalität des Selektionsergebnisses durch zusätzlich eingebrachte Join-Verknüpfungen und Filter beeinflussen.

Beispiel: Pfadausdruck mit Inner-Join-Verknüpfung

Der in Listing 3.19 definierte CDS-View `Z_ViewWithPathExprsChngngCards` nutzt den CDS-View `ZI_Product` aus Listing 3.12 als Datenquelle.

```
define view entity Z_ViewWithPathExprsChngngCards
  as select from ZI_Product
  {
    key Product,
    key _Text[*:inner].Language
  }
```

Listing 3.19 CDS-View mit kardinalitätsänderndem Pfadausdruck

Analyse des Beispiels

Über den Pfadausdruck `_Text[...].Language` wird die Projektionsliste des CDS-Views `Z_ViewWithPathExprsChngngCards` mit dem Sprachschlüssel `Language` des Ziel-CDS-Views `ZI_ProductText` angereichert. Da dieser Ziel-View potenziell Texte in mehreren Sprachen für ein einzelnes Produkt bereitstellen kann, führt die Aufnahme des Sprachschlüssels zu einem Ausmultiplizieren der Datensätze des CDS-Views `ZI_Product` in der Ergebnisliste des CDS-Views `Z_ViewWithPathExprsChngngCards`. Dieser Sachverhalt wird durch die Kardinalitätsangabe `*` im Pfadausdruck signalisiert.

Inner-Join-Verknüpfungen in Pfadausdrücken

Im vorliegenden Fall wird eine Inner-Join- statt einer Left-Outer-Join-Verknüpfung angewendet. Dies wird explizit durch das Sprachelement `INNER` ausgedrückt. Listing 3.20 illustriert die resultierende Create-Anweisung für den Datenbank-View, der aus dem CDS-View `Z_ViewWithPathExprsChngngCards` aus Listing 3.19 generiert wird.

```
CREATE OR REPLACE VIEW "Z_VIEWWITHPATHEXPRSCHNGNGCARDS" AS SELECT
  "ZI_PRODUCT"."MANDT" AS "MANDT",
  "ZI_PRODUCT"."PRODUCT",
  "=AO"."LANGUAGE"
FROM "ZI_PRODUCT" "ZI_PRODUCT" INNER JOIN "ZI_PRODUCTTEXT" "=AO" ON (
  "ZI_PRODUCT"."MANDT" = "=AO"."MANDT" AND
  "=AO"."PRODUCT" = "ZI_PRODUCT"."PRODUCT"
)
WHERE "ZI_PRODUCT"."MANDT" = SESSION_CONTEXT(
  'CDS_CLIENT'
```

Listing 3.20 Create-Anweisung zum Anlegen des Datenbank-Views zum CDS-View aus Listing 3.19

Diese Verknüpfung reduziert die resultierten Datensätze entsprechend der Inner-Join-Semantik. Das heißt, sie nimmt ebenfalls Einfluss auf die Kardinalität der Ergebnisliste des CDS-Views `Z_ViewWithPathExprsChngngCards`.

Wenn Sie Assoziationen in Pfadausdrücken verwenden, können Sie innerhalb dieser Pfade zusätzliche Filterkriterien definieren. Ein Beispiel sehen Sie in Listing 3.21.

Beispiel: Filterkriterien in Pfadausdrücken

```
define view entity Z_ViewWithPathExprsWithFilters
  as select from ZI_Product
{
  key Product,
  _Text[1:Language='E'].ProductName as ProductNameInEnglish,
  _Text[1:Language='E'].Product      as ProductOfEnglishText
}
```

Listing 3.21 CDS-View mit Filtereinschränkungen in Pfadausdrücken

Darin verwendet der CDS-View `Z_ViewWithPathExprsWithFilters` die Assoziation `_Text` des CDS-Views `ZI_Product` aus Listing 3.12, um die beiden Felder `ProductName` und `Product` des Ziel-Views `ZI_ProductText` aus Listing 3.13 in seine Projektionsliste zu übernehmen. Die assoziierten Zieldatensätze werden jedoch auf die Datensätze eingeschränkt, deren Sprachschlüssel `Language` dem konstanten Wert »E« entspricht. Damit wird die in der On-Bedingung der Assoziation `_Text` erfasste Bindung des Felds `Product` des Ziel-CDS-Views um eine weitere Einschränkung ergänzt. Mithin ist der aus den Feldern `Product` und `Language` aufgebaute Schlüssel des Ziel-CDS-Views `ZI_ProductText` vollständig spezifiziert. Die maximale Kardinalität der Verknüpfung der Datensätze über das Pfadsegment sinkt dementsprechend auf »1«. Diese Information wird der Filterbedingung in den beiden skizzierten Pfadausdrücken vorangestellt und von dieser durch einen Doppelpunkt separiert.

Analyse des Beispiels

Da die aus beiden Pfadausdrücken resultierenden Join-Verknüpfungen augenscheinlich identisch sind, sollte der aus dem CDS-View `Z_ViewWithPathExprsWithFilters` abgeleitete Datenbank-View idealerweise nur einen Join enthalten. Tatsächlich findet im vorliegenden Fall eine entsprechende Kondensierung der Join-Operationen statt. In der in Listing 3.22 dargestellten Create-Anweisung zur Anlage des Datenbank-Views findet sich dementsprechend nur ein einzelner Join.

Gleichartige Pfadausdrücke im generierten Datenbank-View

```
CREATE OR REPLACE VIEW "Z_VIEWWITHPATHEXPRSWITHFILTERS" AS SELECT
  "ZI_PRODUCT"."MANDT" AS "MANDT",
  "ZI_PRODUCT"."PRODUCT",
  "=A0"."PRODUCTNAME" AS "PRODUCTNAMEINENGLISH",
  "=A0"."PRODUCT" AS "PRODUCTOFENGLISHTEXT"
FROM "ZI_PRODUCT" "ZI_PRODUCT" LEFT OUTER MANY TO ONE JOIN
  "ZI_PRODUCTTEXT" "=A0" ON (
```

```

"=AO"."PRODUCT" = "ZI_PRODUCT"."PRODUCT" AND
"=AO"."LANGUAGE" = N'E' AND
"ZI_PRODUCT"."MANDT" = "=AO"."MANDT"
)
WHERE "ZI_PRODUCT"."MANDT" = SESSION_CONTEXT(
  'CDS_CLIENT'
)

```

Listing 3.22 Create-Anweisung des Datenbank-Views zum CDS-View aus Listing 3.21

Assoziationen mit Standardfilter

Die Assoziationsdefinition kann auch mit einem Standardfilter ausgestattet werden. Dieser Standardfilter wird angewendet, wenn Sie die Assoziation in Pfadausdrücken verwenden, ohne darin einen neuen Filter zu definieren. Wenn Sie stattdessen einen Filter explizit definieren, wird der Standardfilter ignoriert.

Beispiel: Assoziationsdefinition mit Standardfilter

Im CDS-View `Z_ViewWithAscsWithDfltFilters` aus Listing 3.23 wird ein Beispiel für einen Standardfilter gezeigt.

```

define view entity Z_ViewWithAscsWithDfltFilters
  as select from ZI_Product
  association [0..*] to ZI_ProductText as _TextWithDefaultFilter
    on $projection.Product = _TextWithDefaultFilter.Product
    with default filter _TextWithDefaultFilter.Language = 'E'
  association [0..*] to ZI_ProductText as _Text
    on $projection.Product = _Text.Product
  association [0..1] to ZI_ProductText as _TextInEnglish
    on $projection.Product = _TextInEnglish.Product
    and _TextInEnglish.Language = 'E'
{
  Product,
  _TextWithDefaultFilter.ProductName as ProductNameInEnglish,
  _Text[1:Language='E'].ProductName as ProductNameInEnglish2,
  _TextInEnglish.ProductName as ProductNameInEnglish3,
  _TextWithDefaultFilter[Language='D'].ProductName
    as ProductNameInGerman,
  _TextWithDefaultFilter[*:left outer].ProductName as ProductName,
  _TextWithDefaultFilter,
  _Text,
  _TextInEnglish
}

```

Listing 3.23 Assoziation mit Standardfilter in einem CDS-View

Die Assoziation `_TextWithDefaultFilter` definiert den Standardfilter »E« für das Feld `Language` des Ziel-CDS-Views `ZI_ProductText`. Der Pfadausdruck `_TextWithDefaultFilter.ProductName` liefert demzufolge den englischen Namen `ProductName` des Produkts. Dies entspricht dem Ergebnis der beiden Pfadausdrücke `_Text[1:Language='E'].ProductName` und `_TextInEnglish.ProductName`. Innerhalb des ersten Pfadausdrucks wird dabei der gleiche Filter explizit definiert. Im zweiten Pfadausdruck ist die entsprechende Filterbedingung in der On-Bedingung der Assoziation `_TextInEnglish` verankert.

Analyse des
Beispiels

Der Pfadausdruck `_TextWithDefaultFilter[Language='D']` wendet für das Feld `Language` explizit den Filterwert »D« an. Er liefert den deutschen Produktnamen. Der Standardfilter wird in diesem Fall ignoriert. Wenn Sie den Filter wie im Pfadausdruck `_TextWithDefaultFilter[*:left outer].ProductName` übersteuern, dann werden die Namen sogar in allen Sprachen zurückgeliefert.

Da der Standardfilter nicht immer angewendet wird, ist die maximale Kardinalität der Assoziation `_TextWithDefaultFilter` in Listing 3.23 mit »*« und nicht mit »1« angegeben.

Standardfilter von Assoziationen

Die Standardfilter von Assoziationen wirken nur dann, wenn bei der Verwendung der Assoziationen keine anderen Filter angegeben werden. In der Regel können Filter in Pfadausdrücken die Kardinalität der Assoziationsziele nur reduzieren. Im Gegensatz dazu können übersteuerte Standardfilter die Kardinalität der Assoziationsziele sogar erhöhen. Besonders dann, wenn die Konsumenten einer Assoziation die für diese Assoziation festgelegte Standardfilterung nicht kennen, können sie die zugehörigen Datensätze und damit auch die effektive Kardinalität einer solchen Assoziation nicht vorher-sagen.

Im Zweifelsfall sollten Sie daher eine zusätzliche Assoziation definieren, die explizit die gewünschten Filterkriterien in einer On-Bedingung verankert, anstatt die Standardfilterlogik zu verwenden.



3.6.4 Berechnete Felder in Assoziationsdefinitionen

On-Bedingungen von Assoziationen, die nicht nur exponiert, sondern auch innerhalb des sie definierenden CDS-Views verwendet werden, dürfen keine lokal im CDS-View berechneten Felder referenzieren.

Beispiel:
Referenzierung
berechneter Felder
in Assoziations-
definitionen

Betrachten Sie dazu das Beispiel aus Listing 3.24.

```
define view entity Z_ViewWithAscsWithCalcFields
as select from ZI_SalesOrderItem
association [0..1] to ZI_Product as _Product
    on ZI_SalesOrderItem.Product = _Product.Product
association [0..1] to ZI_Product as _ProductWithAlias
    on $projection.ProductWithAlias = _ProductWithAlias.Product
association [0..1] to ZI_Product as _ProductWithCast
    on $projection.ProductWithCast = _ProductWithCast.Product
association [0..1] to ZI_Product as _ProductWithCastPreservingType
    on $projection.ProductWithCastPreservingType =
        _ProductWithCastPreservingType.Product
{
    key SalesOrder,
    key SalesOrderItem,
    Product,
    _Product,
    _Product.ProductType,
    Product                                as ProductWithAlias,
    _ProductWithAlias,
    _ProductWithAlias.ProductType        as ProductTypeWithAlias,
    cast (Product as matnr)                as ProductWithCast,
    _ProductWithCast,
    cast (Product as matnr preserving type)
        as ProductWithCastPreservingType,
    _ProductWithCastPreservingType,
    _ProductWithCastPreservingType.ProductType
        as ProdTypeWithCastPreservingType
}
```

Listing 3.24 CDS-View mit Assoziationen, deren Definitionen berechnete Felder nutzen

Analyse des
Beispiels

Die Assoziation `_Product` nutzt in ihrer Definition das Feld `Product` der zugrunde liegenden Datenquelle `ZI_SalesOrderItem`. Diese Assoziation kann sowohl exponiert als auch zur Bildung von Pfadausdrücken wie `_Product.ProductType` innerhalb des CDS-Views `Z_ViewWithAscsWithCalcFields` aus Listing 3.24 selbst genutzt werden.

Gleiches gilt für die Assoziation `_ProductWithAlias`, in deren Definition das Feld `ProductWithAlias` einfließt, das über einen Alias aus dem Feld `Product` der zugrunde liegenden Datenquelle `ZI_SalesOrderItem` hervorgeht.

Das Feld `ProductWithCast` wird aus dem Feld `Product` per Cast-Operation berechnet. Es geht in die Definition der Assoziation `_ProductWithCast` ein. Diese Assoziation kann auch exponiert werden. Allerdings ist es nicht zulässig, im CDS-View `Z_ViewWithAscsWithCalcFields` einen Pfadausdruck der Form `_ProductWithCast.ProductType` zu nutzen. Dies ist nur in den verwendeten CDS-Views gestattet.

Beim Feld `ProductWithCastPreservingType` bleibt der technische Datentyp bei der Cast-Operation auf das Datenelement `MATNR` erhalten. Dieser Sachverhalt wird durch den Zusatz `PRESERVING TYPE` ausgedrückt. Wird ein derartig umgewandeltes Feld in die On-Bedingung einer Assoziation eingebracht, darf die Assoziation auch innerhalb des Views selbst zur Bildung von Pfadausdrücken, wie z. B. `_ProductWithCastPreservingType.ProductType`, genutzt werden.

**Cast-Operationen
ohne Typänderung**

3.7 Verwendung von Assoziationen in ABAP-Implementierungen

Exponierte Assoziationen eines CDS-Views können Sie auch im Rahmen von Selektionszugriffen aus Ihrer ABAP-Implementierung heraus nutzen. Ähnlich wie innerhalb Ihrer CDS-View-Modellierung können Sie dabei Pfadausdrücke mit den Assoziationen bilden. Über diese Pfade können Sie auf die Felder der jeweiligen Ziel-CDS-Views zugreifen. Anders als in der CDS-View-Modellierung können Sie jedoch keine Assoziation als Element in die Ergebnisliste Ihrer Selektionsanweisung übernehmen.

**Assoziationen in
ABAP**

Wenn Sie auf eine Assoziation in ABAP zugreifen möchten, müssen Sie dieser einen Schrägstrich (`\`) voranstellen. Das Feld des Assoziationsziels wird durch einen Bindestrich (`-`) vom Assoziationsnamen abgegrenzt. Im Fall von Assoziationsketten werden die einzelnen Assoziationen, mit einem Schrägstrich beginnend, aneinandergereiht. Filterbedingungen können Sie in eckigen Klammern (`[...]`) erfassen.

Listing 3.25 illustriert ein Beispiel für eine Datenselektion mit Pfadausdrücken in ABAP.

**Beispiel:
Datenselektion in
ABAP mit
Pfadausdrücken**

```
SELECT \_salesorder-salesordertype,
       \_salesorderitem\_product\_text[ (1) inner :
                                     where language = 'E' ]-productname
FROM   zi_salesorderscheduleline
WHERE  \_salesorderitem\_product-producttype EQ 'FERT'
INTO TABLE @DATA(lt_result).
```

Listing 3.25 Datenselektion in ABAP unter der Verwendung von Pfadausdrücken

Analyse des Beispiels

In der im Beispiel dargestellten Selektionsanweisung wird, ausgehend von dem CDS-View der Kundenauftragseinteilung `ZI_SalesOrderScheduleLine`, das assoziierte Feld `SalesOrderType` des Kundenauftrags-CDS-Views `ZI_SalesOrder` in die Ergebnisliste `lt_result` zurückgestellt. Zusätzlich wird das Feld `ProductName` aus dem CDS-View `ZI_ProductText` in die Ergebnisliste übernommen.

Durch die Pfadrestriktion `[(1) inner : where language = 'E']` werden dabei entsprechend der Inner-Join-Semantik nur die Datensätze selektiert, für die es einen englischen Text (`Language = 'E'`) gibt. Die in runden Klammern erfasste Kardinalitätsangabe »1« signalisiert dabei, dass es bei der Anzahl der Datensätze im Ergebnis keine Ausmultiplizierungseffekte aufgrund des applizierten Pfadausdrucks geben wird.

Durch die angewandte Where-Bedingung mit dem Pfadausdruck `_sales orderitem_product-producttype` werden schließlich nur die Datensätze der Kundenauftragseinteilung berücksichtigt, deren Position ein Produkt vom Typ `ProductType = »FERT«` zugeordnet ist.