

# Dimensionale Modellierung mit dem OWB 11.2

Maren Eschermann  
Trivadis AG  
Zürich

## Schlüsselworte:

Dimensionale Modellierung, Oracle Warehouse Builder, Entwicklung von Data Marts

## Einleitung

Der Oracle Warehouse Builder bietet viel Unterstützung bei der Entwicklung von Data Marts und ihren Ladeprozessen. Historisierte Dimensionen, Faktentabellen, automatische Fehlerbehandlung und eine Zeitdimension mit mehreren Hierarchien sind mit wenig Aufwand zu realisieren, spezielle Mappingoperatoren erlauben eine effiziente, standardisierte Implementierung der Ladeprozesse. Zur Performancesteigerung können Materialisierte Views (sowohl klassische als auch die mit 11.2 eingeführten sog. „cube based MViews“) aus dem OWB heraus erzeugt werden.

Doch wo liegen die Grenzen dieser automatisch generierten Lösungen? Was sind die Gründe, z.B. auf den Einsatz des Dimensionsoperators zu verzichten und stattdessen den Ladeprozess von Hand zu implementieren?

Das vorliegende Manuskript gibt Antworten auf diese und ähnliche Fragen, vermittelt ein gutes Verständnis der Warehouse Builder Features und ihrer Einsatzmöglichkeiten und bietet praktische Hilfestellung bei der Entscheidung, welche der Warehouse Builder Funktionalitäten im Bereich der dimensionalen Modellierung wann am sinnvollsten eingesetzt werden können.

## Grundlagen der dimensionalen Modellierung

Ein dimensionales Modell eines Data Marts besteht aus Dimensionen, Hierarchien, Kennzahlen und Aggregationsvorschriften, wird z.B. mittels eines ADAPT Modells beschrieben und bildet den Schnittpunkt zwischen der Fach- und der IT-Abteilung.

Bei der Abbildung eines dimensionalen Modells auf ein relationales Modell entsteht häufig ein Semantikverlust. Aus einer relationalen Tabelle die Hierarchie einer Dimension herauszulesen ist schwierig bis unmöglich. Um dem zu begegnen sollten die Metadaten für dimensionale Anwendungen erweitert werden. Genau dies geschieht im Oracle Warehouse Builder, wenn Dimensionen und Cubes modelliert werden.

Der Oracle Warehouse Builder erlaubt sowohl eine relationale als auch eine multidimensionale Implementierung eines Data Marts. Im Folgenden wird die Implementierung eines relationalen Marts beschrieben.

## Modellierung einer Dimension

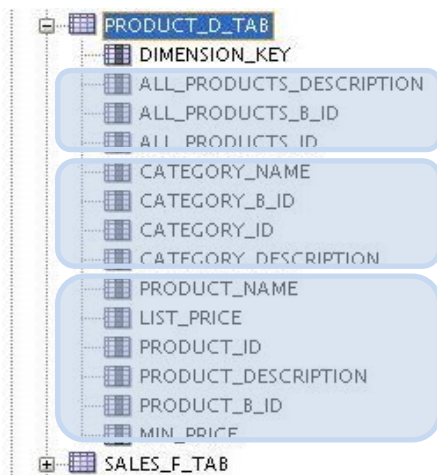
Eine Dimension besteht aus Ebenen (Levels), Hierarchien und Attributen. Jeder Ebene sind eine Anzahl von Attributen zugeordnet, die Beziehungen zwischen Ebenen definiert eine Hierarchie. Eine Dimension kann mehrere Hierarchien haben. Beispielsweise besitzt die Zeitdimen-

### Welcome to the Create Dimension Wizard

This wizard guides you through the creation process.

- Step 1: Provide a name and description.
- Step 2: Set the storage type.
- Step 3: Define the dimension attributes.
- Step 4: Define the levels within the default hierarchy.
- Step 5: Choose the level attributes from the dimension attributes.
- Step 6: Choose the slowly changing dimension type.
- Step 7: Review dimension settings.
- Step 8: Dimension creation progress.

sion häufig eine Wochen-Hierarchie (Tag, Woche) und die Standard-Hierarchie (Tag, Monat, Quartal, Jahr). Mit Hilfe des OWB Dimensions-Wizards erfolgt die Spezifikation in wenigen Schritten. Neben der Dimension werden automatisch eine Tabelle und eine Sequenz als Meta-Objekte im OWB erstellt.



Eine Dimensionstabelle hat folgende Eigenschaften:

- Primärschlüssel auf der Spalte DIMENSION\_KEY
- pro Ebene eine Anzahl von Spalten. Ihre Namen können als Prefix den Namen der Ebene haben.
- pro Ebene einen Index auf den Businesschlüssel

Werden nachträglich Dimensionseigenschaften geändert, so muss ein erneutes „automatisches Binding“ erfolgen, welches jedoch alle manuell vorgenommenen Änderungen an der Tabelle (z.B. Spalten-Kommentare, zusätzliche Constraints) löscht. Ausserdem bekommt der Primärschlüssel einen neuen Namen, was beim Validieren von abhängigen Faktentabellen zu Fehlern führt.

Lassen sich diese Randbedingungen nicht mit projektspezifischen Vorgaben vereinbaren, so kann die Kopplung

zwischen Dimensions- und Tabellenobjekt aufgelöst („unbind“), die Dimensionstabelle manuell erzeugt und die Zuordnung zwischen Dimensionsattributen und Tabellenspalten manuell vorgenommen werden.

Mit Ausnahme der Zeitdimension realisiert der Warehouse Builder Dimensionen immer als „solved dimensions“, d.h. Dimensionstabellen enthalten explizite Zeilen für höhere Ebenen, welche NULL-Einträge für die Ebenenspalten tieferer Granularität enthaltenen. Dadurch ergibt sich die Möglichkeit, dass Faktentabellen auch Elemente höherer Ebenen referenzieren können. Auf diese Weise können Kennzahlen nicht auf Ebene Produkt sondern auf Ebene Produktkategorie geladen werden.

Ein Dimensionsobjekt besitzt Eigenschaften, welche lediglich Anwendung finden, wenn der Dimensionsoperator als „target operator“ in einem Mapping verwendet wird. Dies betrifft Konfigurationen bzgl. Historisierung und Fehlerbehandlung (Orphan Management). Hierauf wird in Kapitel „Implementierung eines Ladeprozesses für eine Dimension“ näher eingegangen.

Im Gegensatz zu früheren OWB Versionen bietet die Version 11.2. erweiterte Möglichkeiten zur Modellierung von Dimensionen an. So ist es möglich, eine Dimension ohne künstlich erzeugten Schlüssel zu generieren, auf diese Weise lassen sich z.B. degenerierte Dimensionen modellieren.

## Zeitdimension

Praktisch jede Faktentabelle referenziert die Zeitdimension mindestens einmal, häufig erfolgt ihre Partitionierung nach der Zeit. In früheren Versionen des OWBs hatte die Tabelle der Zeitdimension wie andere Dimensionstabellen auch einen künstlich erzeugten Schlüssel, der von der Faktentabelle referenziert wurde. Damit war eine Partitionierung nach der Zeit schwierig. Im OWB 11.2. ist der Primärschlüssel einer Tabelle für eine Zeitdimension ein Datumsfeld, so hat auch die Faktentabelle eine Spalte vom Typ Datum, nach der dann sehr einfach partitioniert werden kann.

## Modellierung eines Würfels

Ein Würfel besteht aus Kennzahlen, Dimensionen und Aggregationsvorschriften, wie die Kennzahlen entlang der Hierarchien zu aggregieren sind. Mit Hilfe des OWB Cube-Wizards wird ein Würfel als OWB Meta-Objekt erzeugt werden. Die ebenfalls

- Step 1: Provide a name and description.
- Step 2: Set storage type for the dimension.
- Step 3: Choose the dimensions of the cube.
- Step 4: Specify the measures in the cube.

automatisch generierte Faktentabelle hat folgende Eigenschaften:

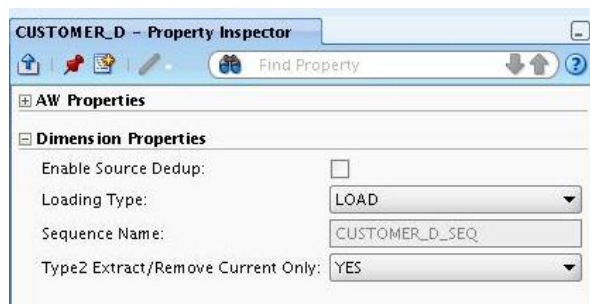
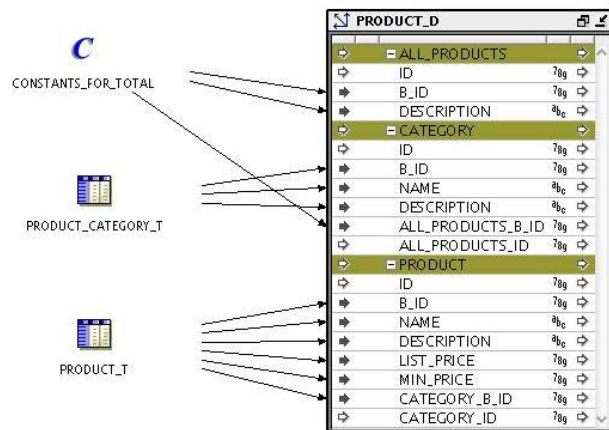
- Tabelle: <NAME CUBE>\_TAB
- Eine Spalte pro Dimension (der Name entspricht im Regelfall dem Namen der Dimension), inkl. FK und optional Bitmap Index
- Eine Spalte pro Kennzahl
- Optional einen zusammengesetzten eindeutigen Schlüssel, bestehend aus der Kombination aller Dimensionsspalten

Wie bei Dimensionen kann auch die Faktentabelle manuell erzeugt und einem Würfel zugeordnet werden. Dies wird in vielen Fällen notwendig, weil projektspezifische Anforderungen insbesondere Performanceanforderungen sich nicht mit den Default-Einstellungen der automatisch generierten Tabelle vereinbaren lassen. So ist die Faktentabelle häufig partitioniert und komprimiert, ihre Foreign Keys werden disabled, um effizienter laden zu können.

### Implementierung eines Ladeprozesses für eine Dimension

Um den ETL Prozess für eine Dimensionstabelle zu implementieren steht der sog. Dimensionsoperator zur Verfügung. Dieser Operator implementiert folgende Funktionalität:

- Erzeugen eines künstlichen Schlüssels
- Lookup der Business Schlüssels
- Deduplizieren von doppelten Elementen
- Historisierung („Slowly Changing Dimensions“)
- Orphan Management



Der Dimensions-Operator erlaubt zwei Ladetypen: *LOAD* und *REMOVE*. Im Allgemeinen wird *LOAD* zum Einsatz kommen, da *REMOVE* Elemente physikalisch löscht. Lediglich bei Verwendung von SCD2 Dimensionen findet der Ladetyp *REMOVE* eine sinnvolle Anwendung (siehe weiter unten).

Ist *Enable Source Dedup* eingeschaltet, was der Defaulteinstellung entspricht, so werden Elemente

auf allen Ebenen dedupliziert. Ist dies auf Grund der Konstellation der Quelldaten nicht notwendig, weil sich z.B. die Quelldaten in Tabellen mit entsprechendem eindeutigem Schlüssel befinden, so kann diese Option ausgeschaltet werden. Werden grosse Datenmengen geladen, kann dies aus Performancegründen ratsam sein, es ist jedoch zu beachten, dass der Dimensionsoperator die Eindeutigkeit der Businessschlüssel in dem Fall nicht gewährleistet.

Beim Ausbringen von dimensionalen Mappings wird pro Dimensionsebene eine OWB\$TEMP Tabelle erzeugt. Das Erzeugen dieser Tabellen lässt sich nicht ausschalten, sie werden intern als Zwischenschritt beim Laden der Dimensionstabelle benötigt.

## Slowly Changing Dimensions

Type 2 slowly changing dimension			
Select attributes to trigger the history saving:			
Levels	Identifying Attribute	Data Type	Record History
[-] ALL_CUSTOMERS			
[-] CUSTOMER			
	BISDAT	DATE	Expiration date
	B_ID	NUMBER	
	DATE_OF_BIRTH	DATE	
	FIRST_NAME	VARCHAR2	
	GENDER	VARCHAR2	
	ID	NUMBER	
	INCOME_LEVEL	VARCHAR2	
	LAST_NAME	VARCHAR2	
	MARITAL_STATUS	VARCHAR2	Trigger history
	VONDAT	DATE	Effective date
	INCOME_LEVEL_ID	NUMBER	Trigger history
[+] INCOME_LEVEL			

Häufig besteht die Anforderung eine Dimension zu historisieren (Slowly Changing Dimension, SCD), wobei meistens der Typ SCD2 (Historisierung aller Versionen) zum Einsatz kommt. Eine solche Dimension benötigt zwei zusätzliche Datumsattribute, welche die Gültigkeit einer Version definieren: *EFFECTIVE\_DATE* definiert das „gültig ab“ einer Version und *EXPIRATION\_DATE* das „gültig bis“. Der Benutzer kann definieren, welche Attribute das Anlegen einer neuen Version auslösen (*Trigger history*) und welche Attribute einfach überschrieben werden. Der OWB bietet die Möglichkeit des sogenannten *Hierarchy Versioning*. Während Kimball die Historisierung nur für Elemente der untersten Ebene vorsieht, erlaubt der OWB auch höhere Ebenen zu

historisieren.

Weitere Einstellungen können bei den Eigenschaften des Dimensionsoperator vorgenommen werden, z.B. wie die Werte für die Datumsfelder gesetzt werden. Das *logische* Löschen einer Version erfolgt über das Setzen des *EXPIRATION\_DATE* auf ein aktuelles Datum. Dieses Verhalten kann durch Einsatz des Ladetyps *REMOVE* und der Einstellung *Type2 Extract/Remove Current Only = Yes* erreicht werden. Die höheren Ebenen dieses Dimensionsoperator dürfen in dem Fall nicht verbunden werden, da diese sonst physikalisch gelöscht würden.

Gibt es zwischen zwei Ladeläufen mehr als eine Änderung auf den Quelldaten, so kann mit der Einstellung *Support Multiple History Loading* geladen werden. Hier werden in einem Ladelauf mehrere Versionen für den gleichen Businessschlüssel erzeugt.

Sollen auch noch nachträglich Versionen geladen werden, die eine frühere Gültigkeit haben als die aktuelle Version, so erlaubt dies die Einstellung *Support Out of Order History Loading*.

Beide Optionen bringen einen Performance-Overhead mit sich, sie sind standardmässig ausgeschaltet.

History Logging Policies	
Default Effective Time of Initial Record:	SYSDATE ...
Default Effective Time of Open Record:	SYSDATE ...
Default Expiration Time of Open Record:	NULL ...
Slowly Changing Type:	Type 2
Support Multiple History Loading:	<input type="checkbox"/>
Support Out of Order History Loading:	<input type="checkbox"/>
Type2 Gap:	1
Type2 Gap Units:	SECONDS

## Fehlerbehandlung (*Orphan Management*)

Orphan management for loading:		
Null parent key values:	Invalid parent key values:	Default Level Row:
<input type="radio"/> No Maintenance	<input type="radio"/> No Maintenance	Settings...
<input checked="" type="radio"/> Default Parent	<input type="radio"/> Default Parent	
<input type="radio"/> Reject Orphan	<input checked="" type="radio"/> Reject Orphan	

Ein mögliches Problem beim Laden von Dimensionen ist, dass für Elemente einer Ebene das Element der höheren Ebene fehlt bzw. nicht spezifiziert ist. Solche Records werden Orphans (Waisen) genannt. Wird versucht ein Produkt zu laden, obwohl die

entsprechende Produktkategorie in der Dimensionstabelle fehlt (*invalid dimension key*), so gibt es für das Ladeverhalten die drei Optionen *No Maintenance*, *Default Parent* und *Reject Orphan*.

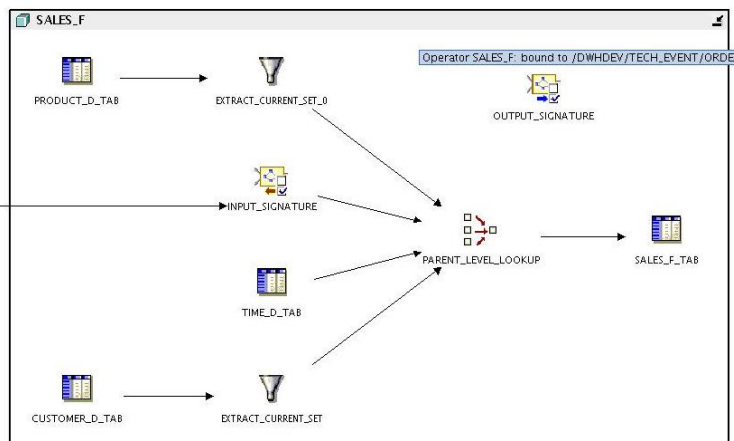
Im ersten Fall werden die fehlerhaften Records geladen und die Attribute auf einen Default Wert gesetzt (in früheren OWB Versionen wurden diese Records herausgefiltert und **nicht** geladen). Die Option *Reject Orphan* lädt solche Records nicht, die zurückgewiesenen Records werden jedoch in

einer Fehlertabelle protokolliert. Damit ergibt sich die Möglichkeit diese Records zu korrigieren und zu einem späteren Zeitpunkt nochmals zu laden.

Anforderungen an die nachgelagerten Reports bringen es häufig mit sich, dass Daten in jedem Fall geladen werden müssen. Diese Datensätze können einem *Default Parent* zugeordnet werden, gleichzeitig werden auch bei dieser Option Datensätze in einer Fehlertabelle protokolliert. Die Attributwerte dieses Default Parents können ebenfalls spezifiziert werden.

Der Dimensionsoperator bietet eine Fülle von Möglichkeiten, wie eine Dimension geladen werden kann. Funktionalitäten, die vom Entwickler mit viel Aufwand immer wieder implementiert werden müssten, stehen in standardisierter Form und gleichbleibender Qualität zur Verfügung. Die Implementierung dieses Operators ist für den Benutzer ersichtlich aber nicht modifizierbar. Die Komplexität der Funktionalität wird deutlich an der Anzahl der intern verwendeten Basisoperatoren: Der Operator für eine Dimension mit drei Ebenen besteht aus 28 Basisoperatoren, der einer SCD2 Dimension immerhin schon aus 34. Ist zusätzlich noch eine Fehlerbehandlung implementiert, benötigt der Operator bereits mehr als 54 Basisoperatoren.

### Implementierung eines Ladeprozesses für einen Würfel

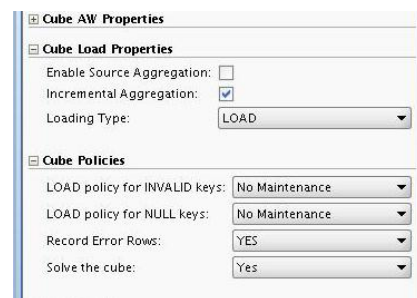


ETL Prozesse für Faktentabellen lassen sich mit dem Cube Operator realisieren. Die Funktionalität besteht aus dem Lookup der Dimensions-Business-Keys (abhängig vom SCD Typ der jeweiligen Dimension) und einer möglichen Fehlerbehandlung. Als Ladetypen stehen *INSERT LOAD*, *LOAD* und *REMOVE* zur Verfügung. *LOAD* erlaubt das Modifizieren bereits geladener Kennzahlen, während *INSERT LOAD* lediglich Kennzahlen hinzufügt, was bei sehr grossen Datenmengen häufig die

einzig mögliche Variante ist.

Das Attribut *ACTIVE\_DATE* bestimmt für referenzierte SCD2 Dimensionen, welches Datum zur Bestimmung der relevanten Dimensionsversion verwendet wird. Wird dieses Attribut nicht verbunden, so wird als Default das aktuelle Datum verwendet.

Auch der Cubeoperator bietet die Möglichkeit zur Fehlerbehandlung, wenn Kennzahlen geladen werden sollen, die eine fehlende oder ungültige Dimensionsreferenz haben. Die Optionen entsprechen denen einer Dimension.



### Erzeugen von materialisierten Views

Der Abfrage-Performance eines Data Mart kommt eine entscheidende Rolle zu. Häufig ist das Aggregieren zum Abfragezeitpunkt zu langsam, so dass mit Hilfe von materialisierten Views eine Voraggregation stattfindet. Der OWB erlaubt beim Ausbringen eines Würfels diese materialisierten Views zu erzeugen, dies ist sowohl für relationale MViews als auch für „cube based“ MViews

möglich. Bei den Eigenschaften eines Würfels kann der Benutzer spezifizieren, welche Ebenen im Voraus berechnet werden sollen.

Die relationalen MViews (pro Dimension und Würfel) sind nicht als OWB Metaobjekte vorhanden, sondern werden lediglich im Targetschema der Datenbank erzeugt. Bei jedem Ausbringen wird hierbei ein neuer Satz von MViews erzeugt. Ein evt. Partitionieren der MView, das Anlegen von MView Logs für ein *Fast Refresh* und das Ermöglichen von Features wie *partition change tracking* müssen manuell vorgenommen werden.

Wählt der Benutzer die Option *ROLAP with MViews* gewählt, so werden beim Ausbringen sowohl der für die „cube based“ MViews notwendige Analytic Workspace, welcher die multidimensionalen Elemente enthält, als auch die entsprechenden MViews erzeugt. Bei der Anwendung dieses Features passiert sehr viel automatisch, ein gutes Verständnis und fundiertes Wissen der verwendeten Technologien ist jedoch notwendig, um mögliche Fehler debuggen und um eine solche Lösung produktiv betreiben zu können.

### **Zusammenfassung**

Die Funktionalitäten, die der OWB bzgl. dimensionaler Modellierung anbietet, sind vielseitig und bieten eine Fülle von Möglichkeiten einen Datamart und vor allem die notwendigen Ladeprozeduren zu implementieren. Die Vorteile hierbei sind, dass das relationale Modell um eine dimensionale Semantik angereichert wird und dass Lademappings effizient und vor allem standardisiert implementiert werden können. Die Realisierung der Historisierung oder eine Fehlerbehandlung sind innerhalb eines Projekts und projektübergreifend einheitlich gelöst. Dies erlaubt nicht nur eine schnelle Realisierung sondern auch das Erreichen einer hohen Qualität in der Umsetzung und eine gute Wartbarkeit des Systems.

Die dimensional Features des Warehouse Builders gleichen einem Werkzeugkasten, aus dem jedes Projektteam sich diejenigen Teile herausgreifen kann, die sich mit den Projektanforderungen vereinbaren lassen. Der Dimensionsoperator kann verwendet werden, während auf den Cubeoperator verzichtet wird, weil hier eine Partition Exchange Strategie zum Einsatz kommen soll, die der Cube Operator (noch) nicht unterstützt. Ein Prototyping zu Beginn eines Projektes kann bei der Entscheidung helfen. Wie auch immer ein Projektteam sich entscheidet, welches Features auch immer zum Einsatz kommen, am Ende sollte eine standardisierte, flexible Lösung stehen, die gut zu warten und zu erweitern ist.

Kontaktadresse:

Maren Eschermann  
Trivadis AG  
Europa Strasse 5  
CH-8153 Glattbrugg

Telefon: +41 (0) 44-808 7020  
Fax: +41 (0) 44-808 7021  
E-Mail: [maren.eschermann@trivadis.com](mailto:maren.eschermann@trivadis.com)  
Internet: [www.trivadis.com](http://www.trivadis.com)