

# **Star-Schema-Modellierung mit ERwin - eine kritische Reflexion der Leistungspotentiale und Anwendungsmöglichkeiten**

*Michael Hahne*  
*T&I GmbH*

**Workshop MSS-2000**

*Bochum, 24. März 2000*

## *Worum es geht...*

### Data Warehouse-Modellierung

⇒ Modellierung auf logischer und physischer Ebene

⇒ Modellierung für relationale Zielsysteme

⇒ Star-Schema-Modellierung

### *Fragestellung:*

Wie kann durch das Modellierungswerkzeug ERwin

- die *Modellierung* des Data Warehouse-Modells,
- die *Dokumentation* des Data Warehouse,
- die *Implementierung* des Data Warehouse-Modells
- und die spätere *Wartung* des Data Warehouse

unterstützt und erleichtert werden?

# Gliederung

I Sicht des Data Warehouse-Modellierers bzw. Designers  
(*logische/physikalische Modellierungsebene*)

- Abbildung der Strukturen
- Unterstützung durch das Werkzeug

II Sicht des Data Warehouse-Administrators bzw. Implementierers  
(*physikalische Modellierungsebene*)

- Optimierungspotentiale der Zieldatenbanken
- Größenberechnung und -vorhersage

III Dokumentation und Metadaten

IV Fazit

# Gliederung

## I Sicht des Data Warehouse-Modellierers bzw. Designers (*logische/physikalische Modellierungsebene*)

- Abbildung der Strukturen
- Unterstützung durch das Werkzeug

## II Sicht des Data Warehouse-Administrators bzw. Implementierers (*physikalische Modellierungsebene*)

- Optimierungspotentiale der Zieldatenbanken
- Größenberechnung und -vorhersage

## III Dokumentation und Metadaten

## IV Fazit

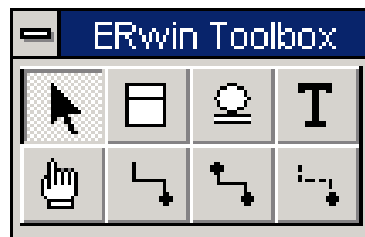
# Modellierungsmethodik in ERwin

IDEF1X

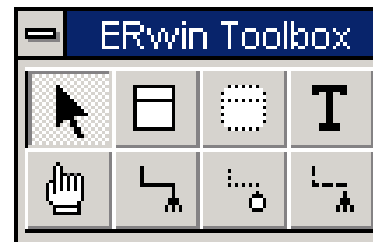
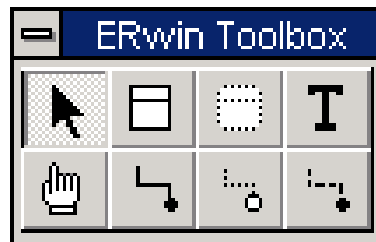
IE

DM

logical



physical

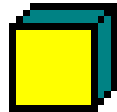


## *Dimensionale Modelle in ERwin*

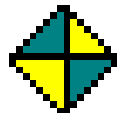
Erwin modelliert im Prinzip ein Star Schema.

Dafür werden erweiterte Funktionen zur Verfügung gestellt.

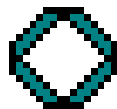
Es werden drei Rollen unterschieden, die eine Tabelle in einem dimensionalen Modell in ERwin wahrnehmen kann:



fact table



dimension table



outrigger table

# Gliederung

I Sicht des Data Warehouse-Modellierers bzw. Designers  
(*logische/physikalische Modellierungsebene*)

- **Abbildung der Strukturen**
- Unterstützung durch das Werkzeug

II Sicht des Data Warehouse-Administrators bzw. Implementierers  
(*physikalische Modellierungsebene*)

- Optimierungspotentiale der Zieldatenbanken
- Größenberechnung und -vorhersage

III Dokumentation und Metadaten

IV Fazit

## *Rollen von Tabellen in dimensionalen Modellen*

Die Rolle, die eine Tabelle in einem dimensionalen Modell einnimmt, kann manuell vergeben werden oder nach den folgenden Regeln automatisch ermittelt werden:

Faktentabelle (*fact table*):

- keine Vaterbeziehung

Dimensionstabelle (*dimension table*):

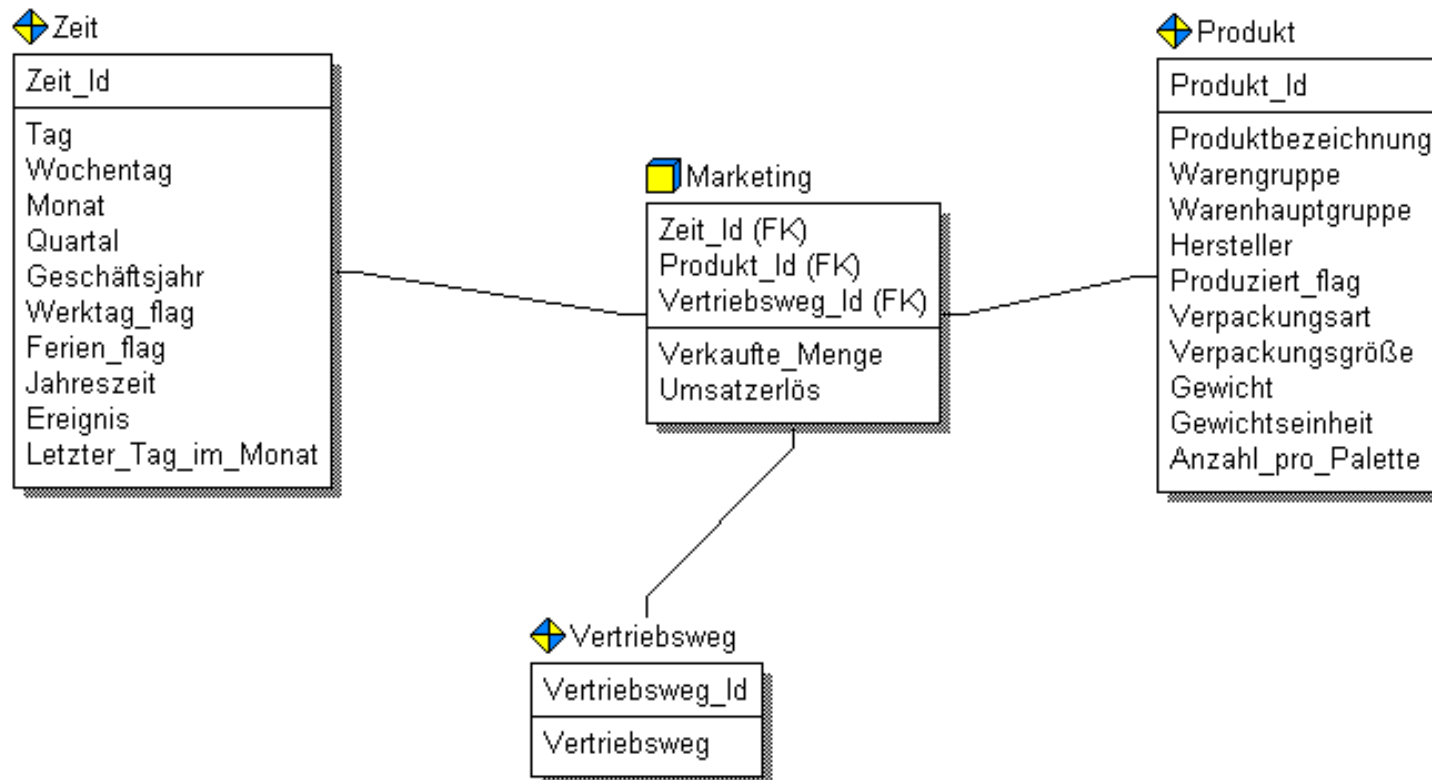
- keine Beziehungen zu irgendeiner Tabelle oder
- eine Vaterbeziehung zu einer Faktentabelle oder
- eine Vater- und eine Kindbeziehung

„snow-flaked“ Dimensionstabelle (*outrigger table*)

- eine Vaterbeziehung zu einer Dimensionstabelle



# Beispielmodell



## *Hierarchien in Dimensionen*

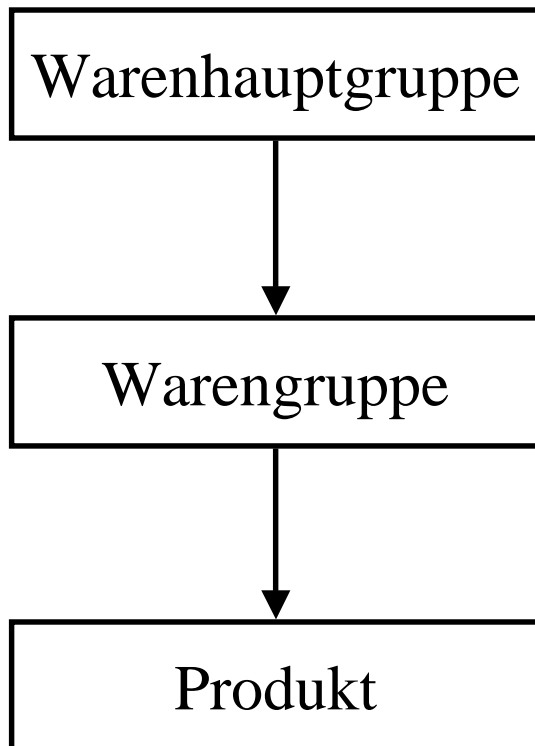
Die Dimensionselemente haben Beziehungen untereinander.  
Die resultierenden Strukturen:

- Flache Strukturen
- Bäume
- Wälder
- Parallele Hierarchien (multiple Verdichtungswege)

Im Star-Schema gibt es zwei grundsätzliche Möglichkeiten der Abbildung von Hierarchien:

- Implizit über Attribute in der Dimensionstabelle
- Explizit über modellierte Beziehungen

# Produktdimension

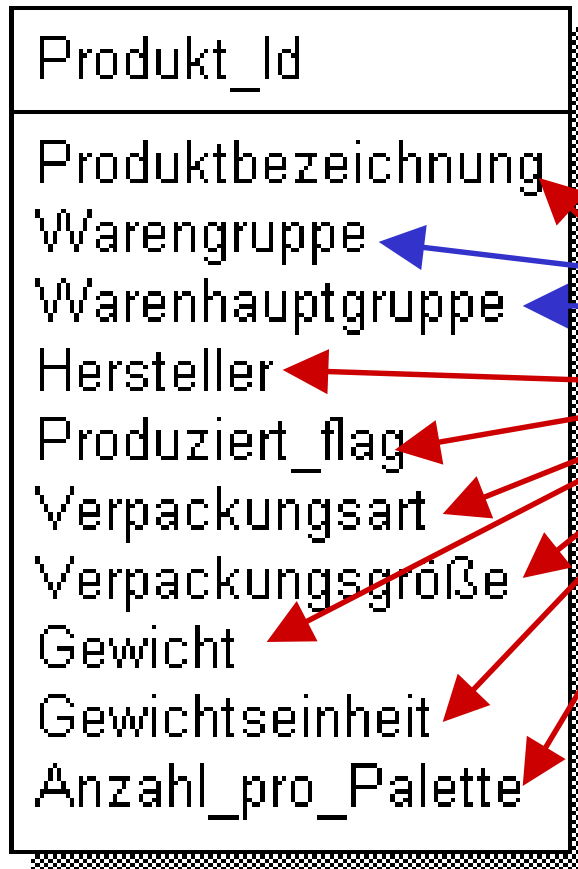


## ◆ Produkt

Produkt_Id
Produktbezeichnung
Warengruppe
Warenhauptgruppe
Hersteller
Produziert_flag
Verpackungsart
Verpackungsgröße
Gewicht
Gewichtseinheit
Anzahl_pro_Palette

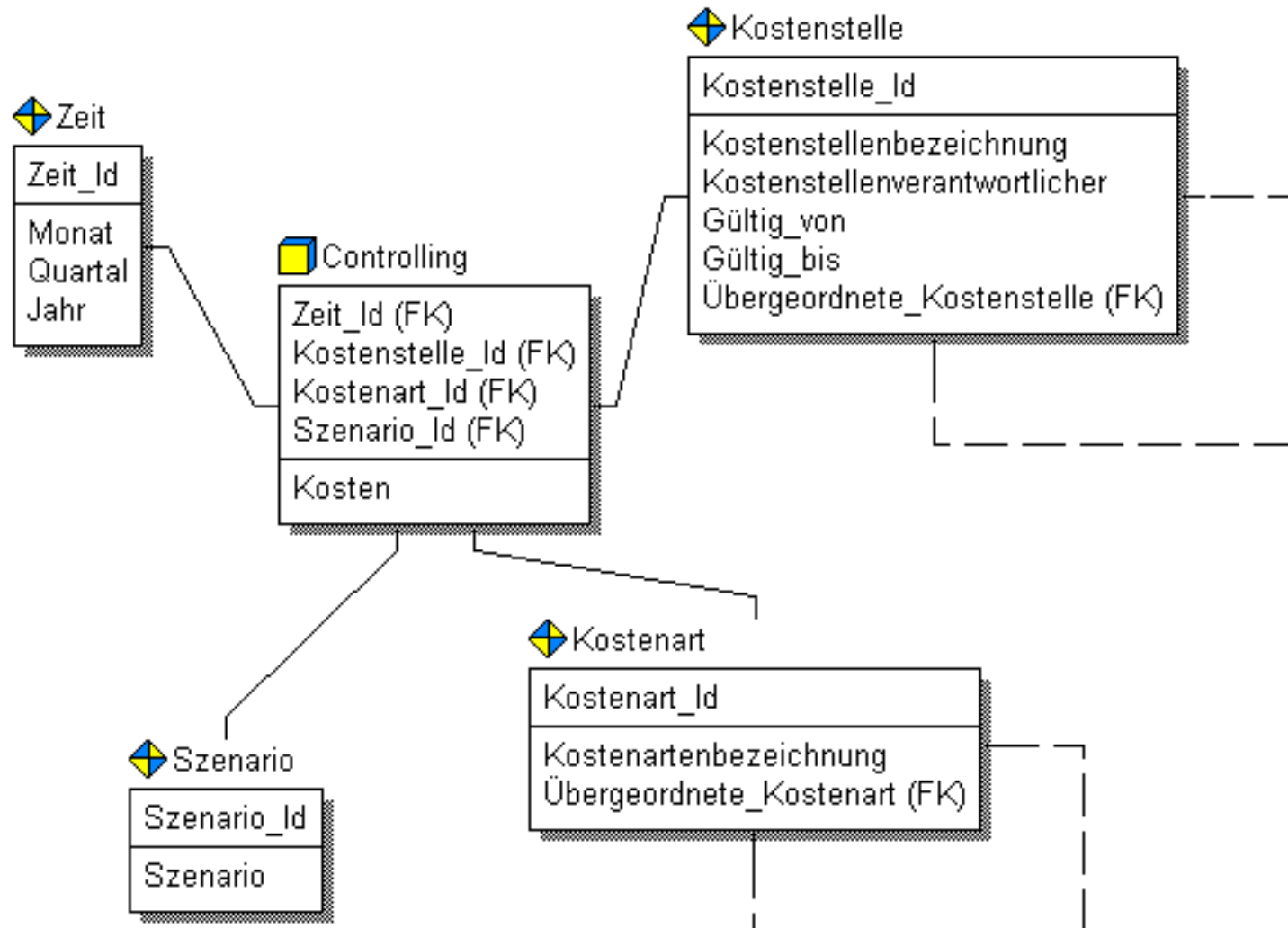
# Hierarchien über Attribute

◆ Produkt

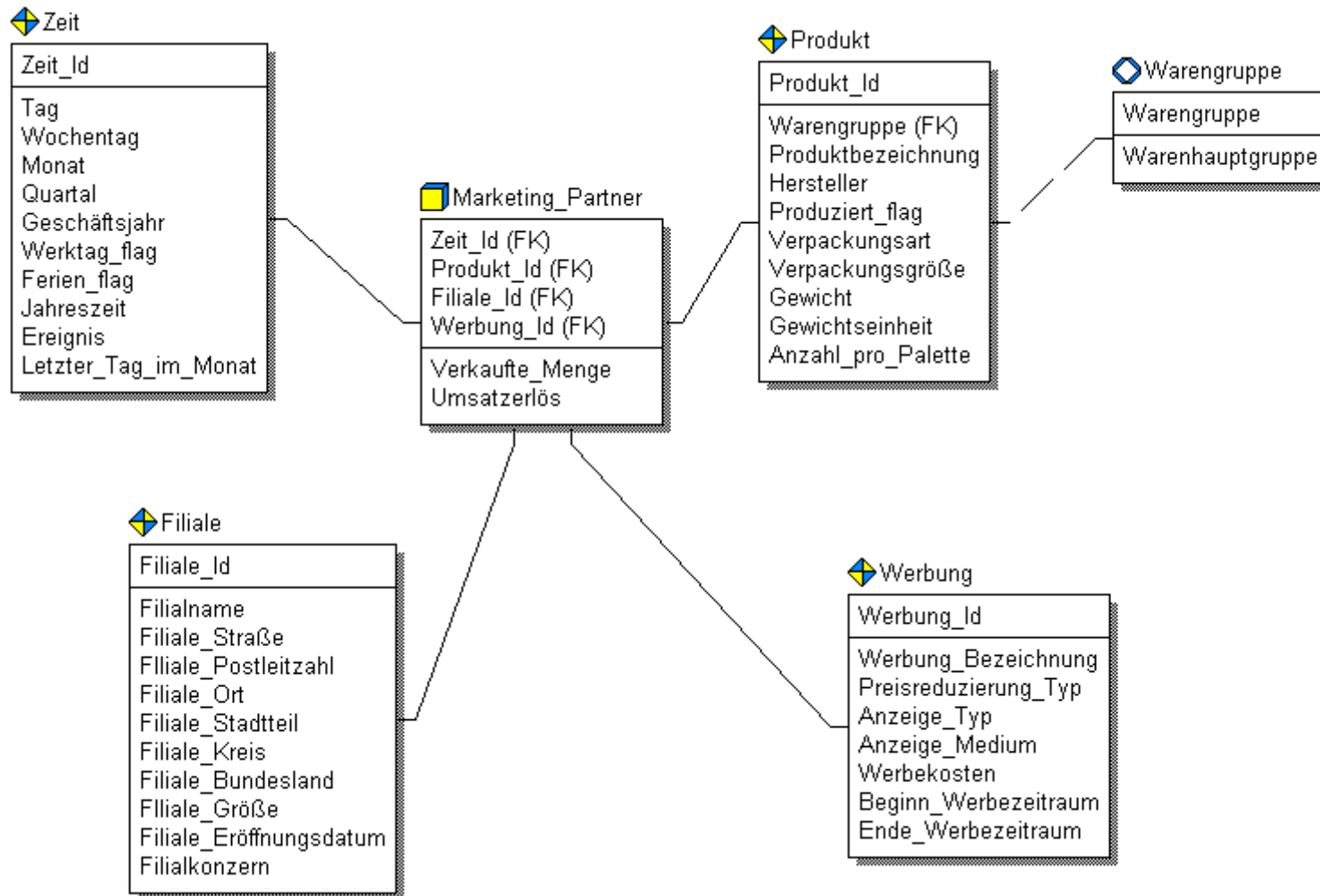


Eine Differenzierung zwischen Hierarchie-Attributen und normalen Attributen ist nicht erkennbar

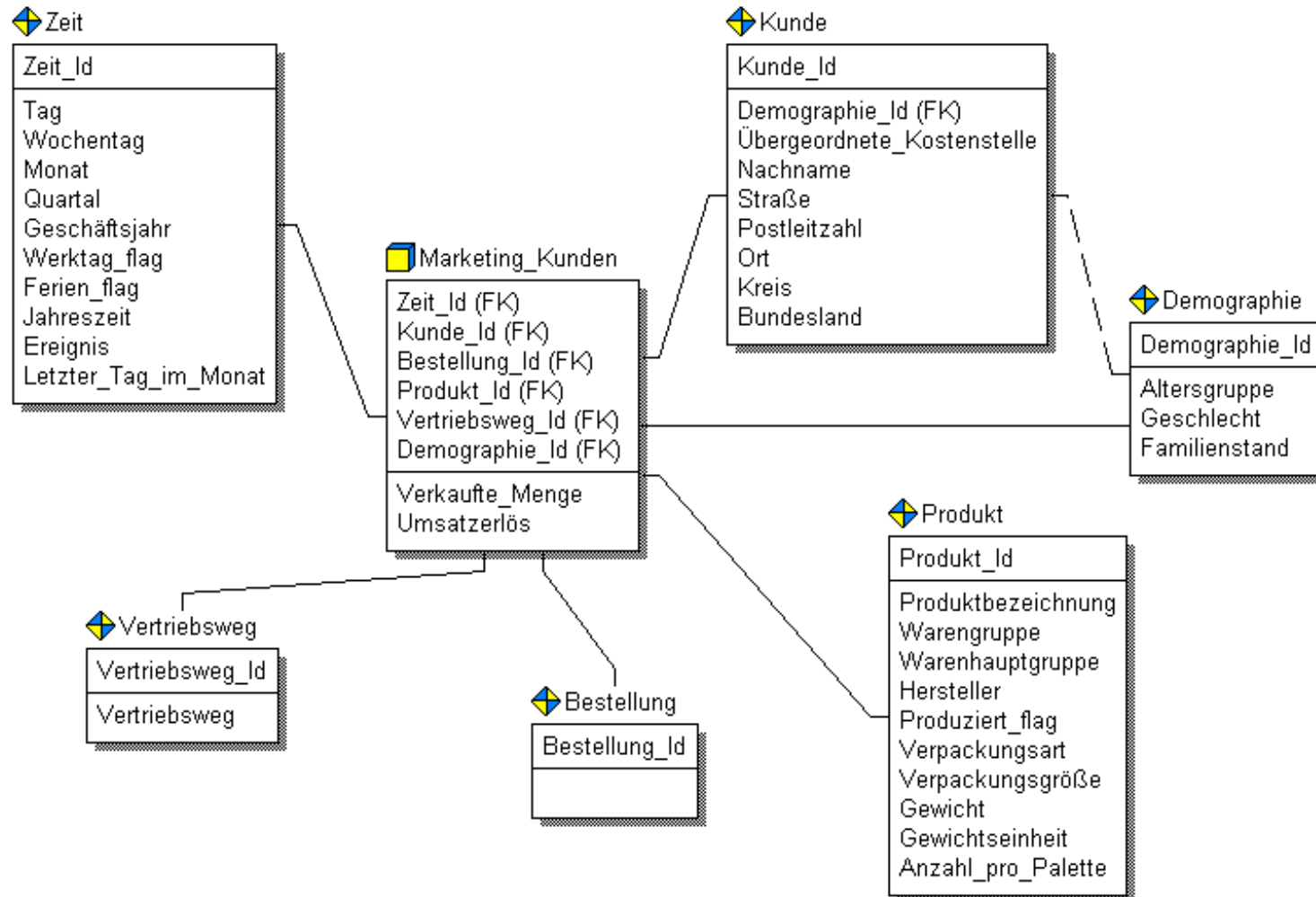
# Hierarchien durch modellierte Beziehungen



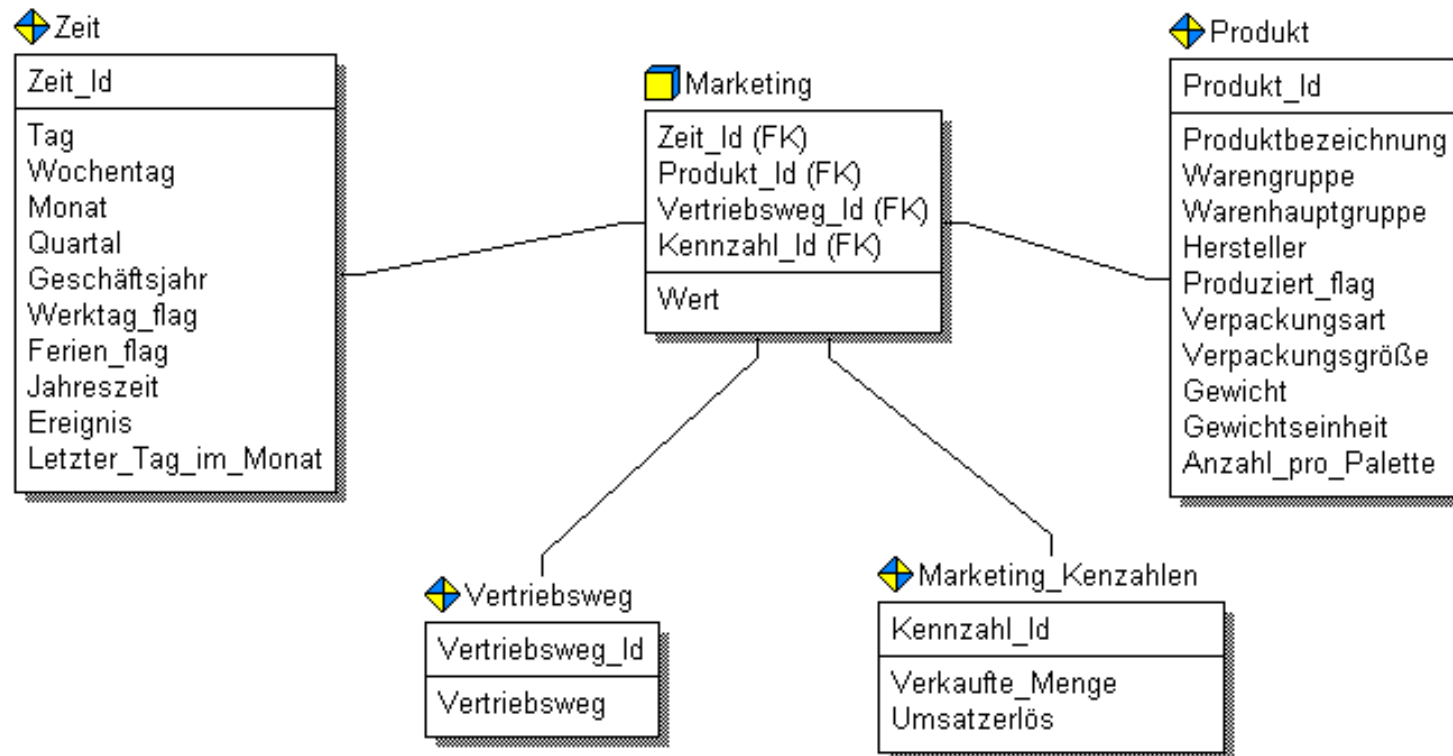
# Snow Flaking



# Minidimension



# Kennzahlen: Faktentabelle vs. Dimension





# Gliederung

I Sicht des Data Warehouse-Modellierers bzw. Designers  
(*logische/physikalische Modellierungsebene*)

- Abbildung der Strukturen
- Unterstützung durch das Werkzeug

II Sicht des Data Warehouse-Administrators bzw. Implementierers  
(*physikalische Modellierungsebene*)

- Optimierungspotentiale der Zieldatenbanken
- Größenberechnung und -vorhersage

III Dokumentation und Metadaten

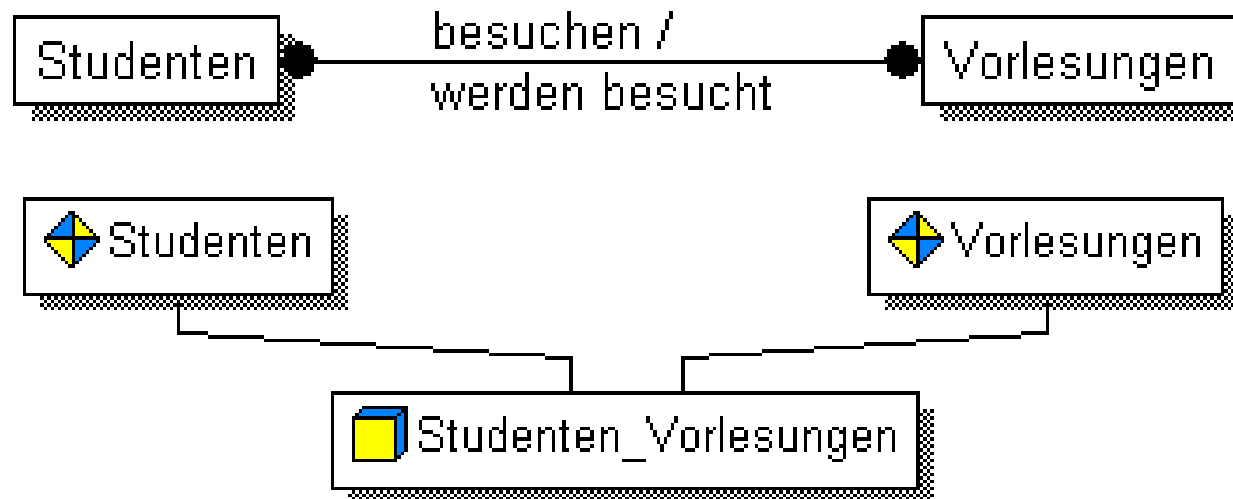
IV Fazit

## *m:n-Beziehungen*

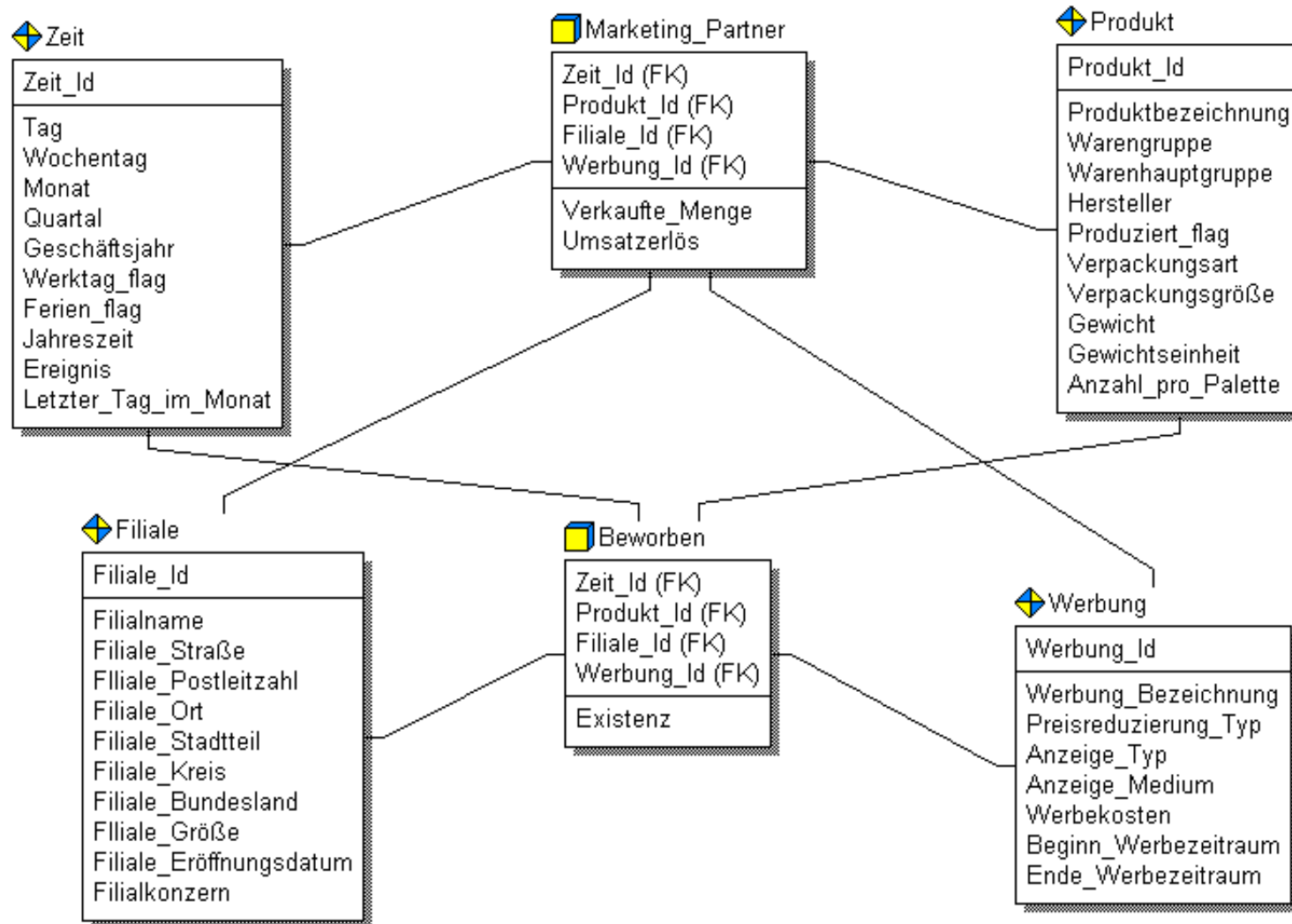
Erwin kann m:n-Beziehungen automatisch auflösen und generiert eine assoziierte Tabelle.

In einem dimensionalen Modell ist diese assoziierte Tabelle eine Faktentabelle

Anwendungsbeispiel: *factless fact table*



# Überlagerungstabelle



## *slowly changing dimensions*

Dimensionstabellen im Star Schema sind zunächst statisch. Im Zeitablauf ändern sich jedoch z. B. Kundendaten etc.

Es gibt drei Arten der Historienverfolgung :

**Typ I:** Der neue Datensatz überschreibt den alten  
(-> *keine Historienverfolgung*)

**Typ II:** Ein zusätzlicher Datensatz wird angelegt  
(*komplette Historienverfolgung, erfordert generalisierten Schlüssel*)

**Typ III:** Neue Felder werden angelegt: originale und aktuelle  
(*keine Zwischenwerte mehr vorhanden*)

Diese Eigenschaft einer Dimensionstabelle wird in ERwin gepflegt und dient der Dokumentation über den Berichts-Generator.

# Gliederung

I Sicht des Data Warehouse-Modellierers bzw. Designers  
(*logische/physikalische Modellierungsebene*)

- Abbildung der Strukturen
- Unterstützung durch das Werkzeug

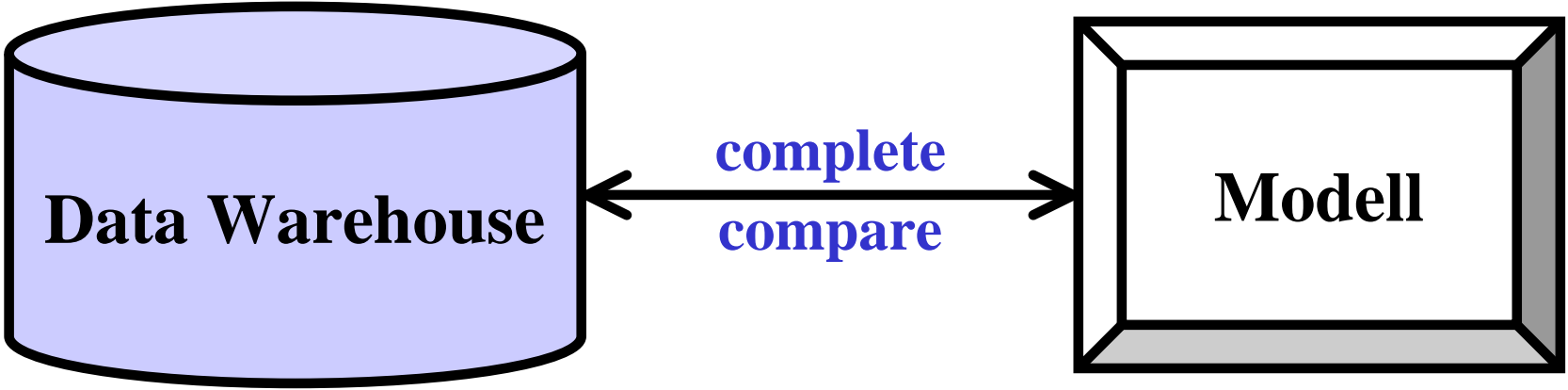
II Sicht des Data Warehouse-Administrators bzw. Implementierers  
(*physikalische Modellierungsebene*)

- Optimierungspotentiale der Zieldatenbanken
- Größenberechnung und -vorhersage

III Dokumentation und Metadaten

IV Fazit

# *Wartung des Modells*



# Gliederung

I Sicht des Data Warehouse-Modellierers bzw. Designers  
(*logische/physikalische Modellierungsebene*)

- Abbildung der Strukturen
- Unterstützung durch das Werkzeug

II Sicht des Data Warehouse-Administrators bzw. Implementierers  
(*physikalische Modellierungsebene*)

- Optimierungspotentiale der Zieldatenbanken
- Größenberechnung und -vorhersage

III Dokumentation und Metadaten

IV Fazit

## *Parallelisierung*

Zur Steigerung der Performance von Data Warehouse-Systemen werden vielfach Mechanismen der parallelen Verarbeitung herangezogen.

Bei Datenbanken ist Platten-I/O der ausschlaggebende Faktor und damit gewinnt das physikalische Design an Bedeutung.

Die Unterstützung parallelen Platten-I/Os wird von Datenbanken z. B. in Form von Partitionierung von Tabellen angeboten.

Diese Features sind über ERwin im physikalischen Modell einstellbar.



# Gliederung

I Sicht des Data Warehouse-Modellierers bzw. Designers  
(*logische/physikalische Modellierungsebene*)

- Abbildung der Strukturen
- Unterstützung durch das Werkzeug

II Sicht des Data Warehouse-Administrators bzw. Implementierers  
(*physikalische Modellierungsebene*)

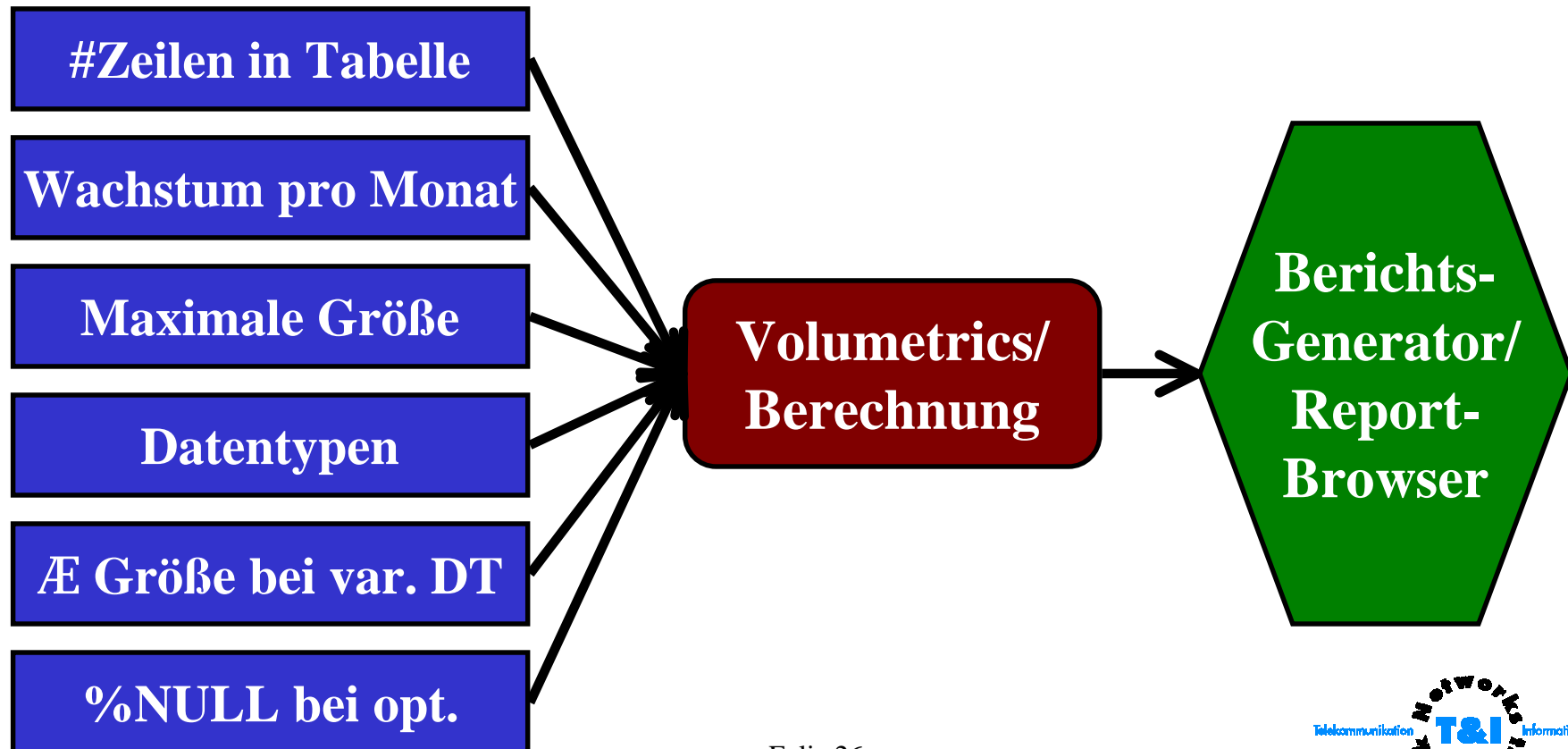
- Optimierungspotentiale der Zieldatenbanken
- Größenberechnung und -vorhersage

III Dokumentation und Metadaten

IV Fazit

## Größenberechnung (*Volumetrics*)

Durch *Volumetrics* wird die Größe von Tabellen, Indizes und physikalischen Datenbank-Speicherobjekten berechnet und über eine Zeitachse vorhergesagt.



# Gliederung

I Sicht des Data Warehouse-Modellierers bzw. Designers  
(*logische/physikalische Modellierungsebene*)

- Abbildung der Strukturen
- Unterstützung durch das Werkzeug

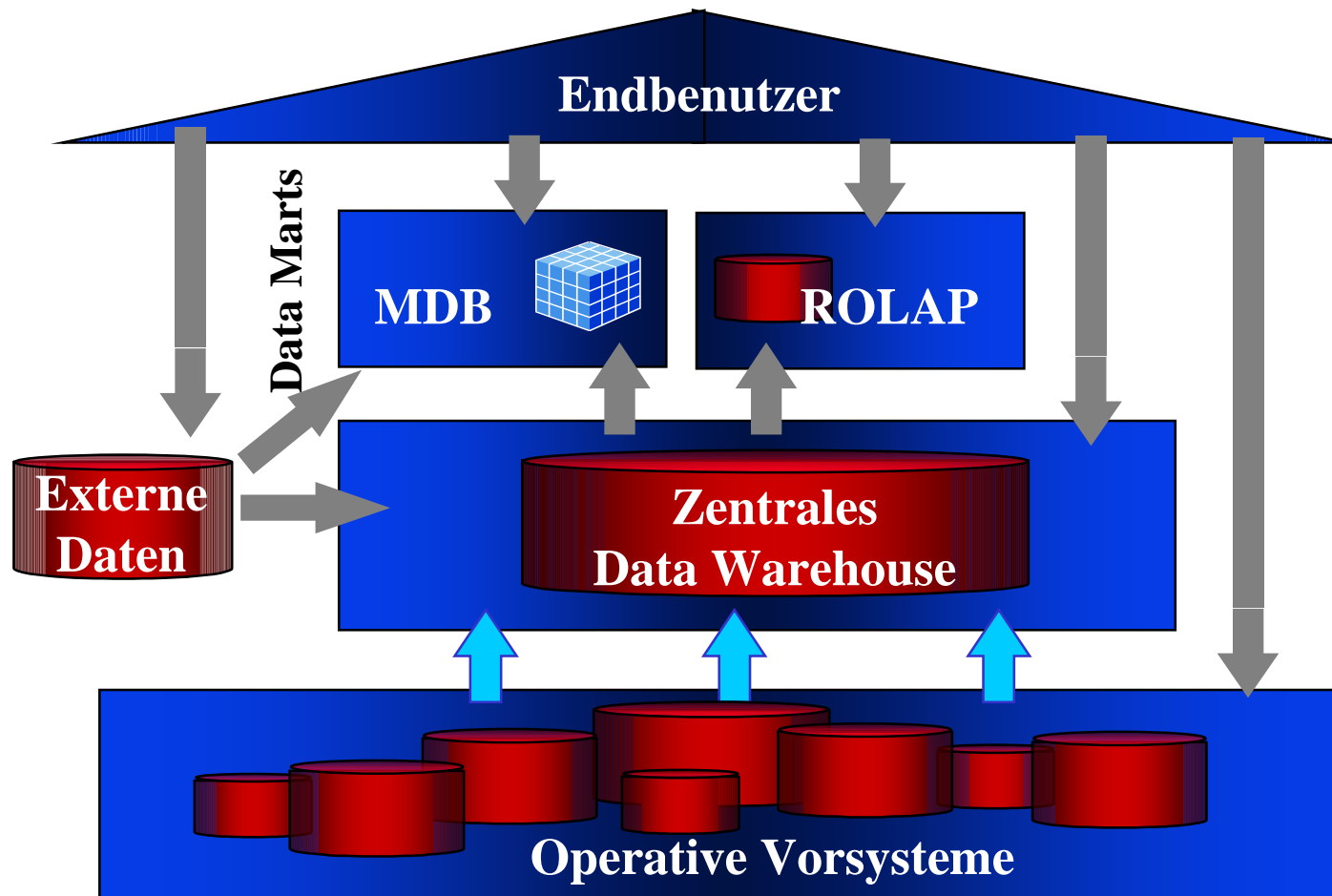
II Sicht des Data Warehouse-Administrators bzw. Implementierers  
(*physikalische Modellierungsebene*)

- Optimierungspotentiale der Zieldatenbanken
- Größenberechnung und -vorhersage

III Dokumentation und Metadaten

IV Fazit

# „Datenpumpe“ innerhalb der Referenzarchitektur



## *Dokumentation von Data Warehouse-Datenquellen*

- Quelldatenbanken zur Dokumentation des Uploads.
- Struktur der Quelldaten kann manuell erfaßt werden
- Über Reverse-Engineering Funktionalität kann die Quelldatenstruktur direkt importiert werden
- Keine Programmfunktionalität zum Datentransfer
- Nur Dokumentationscharakter

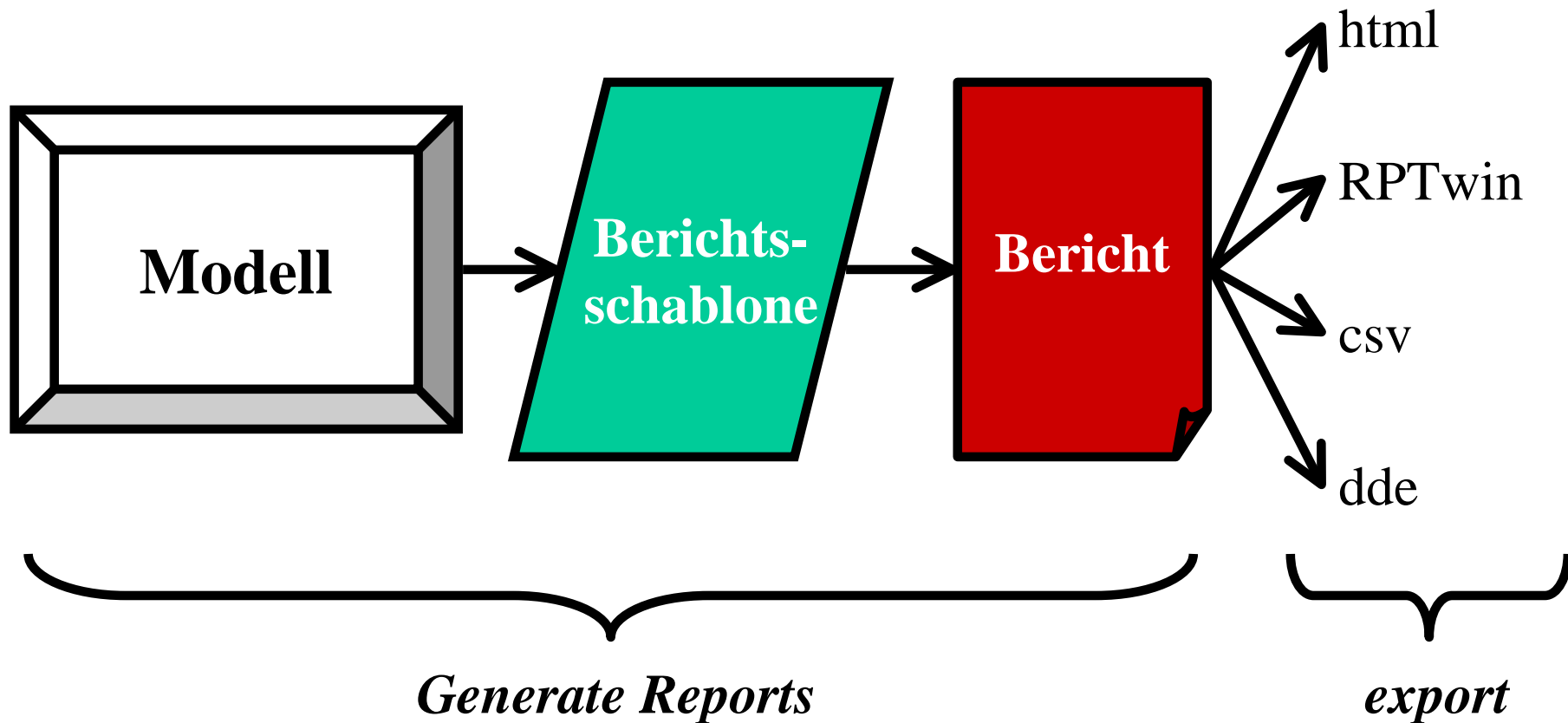
## *Dokumentation von Data Warehouse-Regeln*

Es können Regeln definiert werden, denen die Tabellen eines dimensionalen Modells genügen sollen:

- Refresh-Regeln
- Update-Regeln
- Backup-Regeln
- Recovery-Regeln
- Archiving-Regeln
- Purge-Regeln

Diese Regeln dienen nur der Dokumentation der hinter den Strukturen liegenden Funktionalität zur Wartung eines Data Warehouse.

# *Dokumentation mit Report-Generator*



# Gliederung

I Sicht des Data Warehouse-Modellierers bzw. Designers  
(*logische/physikalische Modellierungsebene*)

- Abbildung der Strukturen
- Unterstützung durch das Werkzeug

II Sicht des Data Warehouse-Administrators bzw. Implementierers  
(*physikalische Modellierungsebene*)

- Optimierungspotentiale der Zieldatenbanken
- Größenberechnung und -vorhersage

III Dokumentation und Metadaten

IV Fazit



## *Fazit*

Durch das Modellierungswerkzeug ERwin kann

- die *Modellierung* des Data Warehouse-Modells,
  - die *Dokumentation* des Data Warehouse,
  - die *Implementierung* des Data Warehouse-Modells
  - und die spätere *Wartung* des Data Warehouse
- unterstützt und erleichtert werden!