

# Haben Sie die Zeit im Griff?

## Designtipps zur Zeitdimension

**Dani Schnider**  
**Trivadis AG**  
**Zürich/Glattbrugg, Schweiz**

### Schlüsselworte:

Data Warehouse, Data Mart, Star Schema, Dimensionale Modellierung, Zeitdimension

### Einleitung

Abfragen im Data Warehouse haben fast immer einen Zeitbezug. Ob es dabei um die Mitarbeiterauslastung im letzten Quartal, den Umsatz pro Monat und Produktkategorie im laufenden Jahr, die Anzahl der Transaktionen der letzten drei Tage, die Auswertung der Online-Bestellungen pro Wochentag und Land oder die Vertragsabschlüsse pro Monat im laufenden Jahr im Vergleich zum Vorjahr geht – eine der Dimensionen in jedem Data Mart ist immer die Zeitdimension.

### Die Zeitdimension im Star Schema

Die Zeitdimension wird üblicherweise wie in Abbildung 1 dargestellt als explizite Dimensionstabelle modelliert und mit allen Attributen versehen, die als Filter- oder Gruppierungskriterien in Auswertungen relevant sein können. Dazu gehören einfache Attribute wie Datum, Wochentag, Monatsbezeichnung, Quartal oder Jahr, teilweise aber auch zusätzliche Informationen, wie wir später sehen werden. Durch eine geeignete Auswahl von Attributen können die Abfragen auf ein Star Schema so vereinfacht werden, dass keine Datumarithmetik notwendig ist, sondern dass ausschließlich vorberechnete Werte für Datumabfragen verwendet werden können.

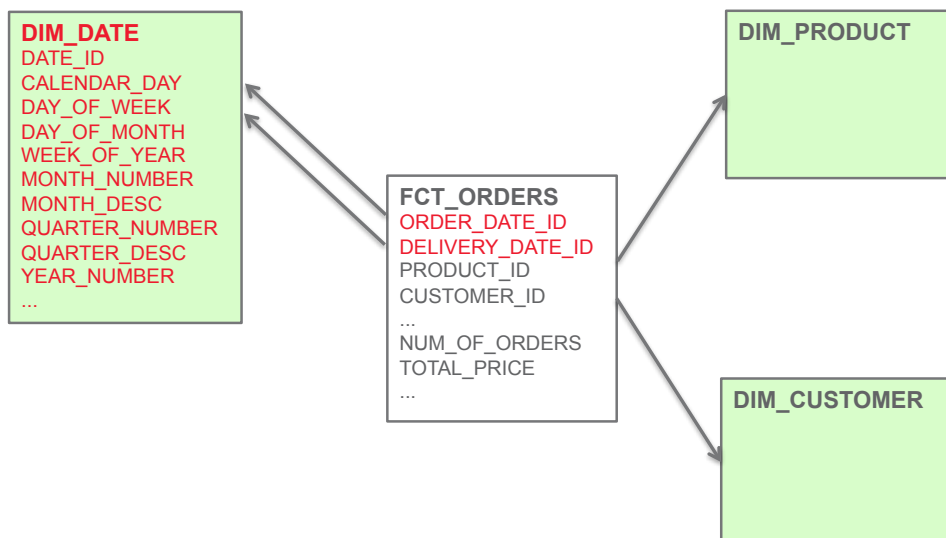


Abb. 1: Star Schema mit Zeitdimension

Das Grundprinzip einer Zeitdimension ist einfach und wird fast in jedem Data Warehouse verwendet. Doch wie sieht es aus mit etwas komplexeren Abfragen, bei denen zum Beispiel Feiertage berücksichtigt werden müssen? Diese Thematik wird nachfolgend anhand eines Beispiels genauer beleuchtet. Das Anwendungsbeispiel ist zwar erfunden, ist aber eine typische Anforderung, wie sie in vielen DWH-Projekten vorkommen kann.

### Wie lange dauert die Auslieferung einer Bestellung?

Der Betreiber eines fiktiven Web-Shops möchte auswerten, wie viele Tage es zwischen der Bestellung eines Produktes und dem Versand an den Kunden dauert. Diese Auswertung ist sehr einfach, wenn in der Faktentabelle sowohl Bestell- als auch Lieferdatum referenziert werden, wie dies in unserem Star Schema (vgl. Abbildung 1) der Fall ist. Mit einer einfachen SQL-Abfrage kann die Anzahl der Tage durch Differenzbildung zwischen den beiden Daten ermittelt werden:

```
SELECT od.calendar_day order_date
      , dd.calendar_day delivery_date
      , dd.calendar_day - od.calendar_day AS num_of_calendar_days
FROM fct_orders f
JOIN dim_date od ON (od.date_id = f.order_date_id)
JOIN dim_date dd ON (dd.date_id = f.delivery_date_id)
```

Das Problem dieser Abfrage ist jedoch, dass sie die Anzahl der Kalendertage zwischen zwei Ereignissen ermittelt. Diese Auswertung ist in vielen Fällen nicht genügend, da sie Wochenenden und Feiertage nicht berücksichtigt. Zwischen dem 30. Dezember 2011 und dem 9. Januar 2012 liegen zwar 10 Kalendertage, jedoch nur 5 Arbeitstage, da in diesem Zeitintervall zwei Wochenenden sowie zwei Feiertage (wovon einer auf ein Wochenende fällt) liegen:

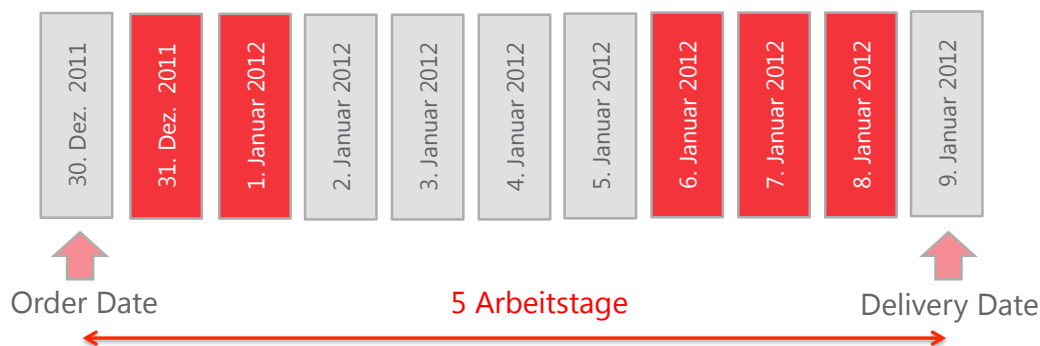


Abb. 2: Anzahl Arbeitstage zwischen 30. Dezember 2011 und 9. Januar 2012

Um die Arbeitstage im Kalender einfach ermitteln zu können, wird ein Abfrage-Indikator eingeführt, welcher für jedes Datum festhält, ob es sich um einen Arbeitstag oder um einen arbeitsfreien Tag (d.h. Samstag, Sonntag oder Feiertag) handelt. Doch wie soll nun mit einer einfachen und effizienten SQL-Abfrage die Anzahl der Arbeitstage ermittelt werden?

Das Berechnen der Anzahl Arbeitstage ist nicht ganz trivial, da nun zusätzlich anhand des Indikators die Arbeitstage gezählt werden müssen. Um solche Abfragen so einfach wie möglich zu halten, wird diese Arbeit bereits beim Füllen der Zeitdimension ausgeführt: Die Arbeitstage werden „durchgezählt“ und als zusätzliches Attribut in der Zeitdimension gespeichert. Zusätzlich zum Indikator wird für jeden Tag die Nummer des Arbeitstages seit Beginn der Zeitrechnung festgehalten. Dabei wird für arbeitsfreie Tage (Wochenenden und Feiertage) jeweils die Nummer des darauf folgenden

Arbeitstages gespeichert. Wenn also ein Kunde an einem Samstag eine Bestellung ausführt, wird diese Bestellung dem darauffolgenden Montag zugeordnet. Für das Datumintervall vom 30. Dezember 2011 bis 9. Januar 2012 stehen beispielsweise folgende Informationen über Arbeitstage in der Zeitdimension:

CALENDAR_DAY	DAY_OF_WEEK	WORKING_DAY_IND	WORKING_DAY_NR
30.12.2011	Friday	1	1516
31.12.2011	Saturday	0	1517
01.01.2012	Sunday	0	1517
02.01.2012	Monday	1	1517
03.01.2012	Tuesday	1	1518
04.01.2012	Wednesday	1	1519
05.01.2012	Thursday	1	1520
06.01.2012	Friday	0	1521
07.01.2012	Saturday	0	1521
08.01.2012	Sunday	0	1521
09.01.2012	Monday	1	1521

Mit Hilfe dieser zusätzlichen Information ist es nun einfach möglich, die Anzahl der Arbeitstage zwischen Bestell- und Lieferdatum zu ermitteln:

```
SELECT od.calendar_day order_date
      , dd.calendar_day delivery_date
      , dd.working_day_num - od.working_day_num AS num_of_working_days
FROM fct_orders f
JOIN dim_date od ON (od.date_id = f.order_date_id)
JOIN dim_date dd ON (dd.date_id = f.delivery_date_id)
```

Die Anzahl der Arbeitstage zwischen dem 30. Dezember 2011 (Arbeitstag 1516) und dem 9. Januar 2012 (Arbeitstag 1521) wird nun korrekt berechnet und beträgt 5 Arbeitstage.

### Regionale Feiertage

Für einen Web-Shop mit einem Standort ist diese Lösung genügend. Aber nehmen wir an, unser fiktiver Web-Shop ist international tätig. Je nachdem, ob ein Kunde sein Produkt über die entsprechende Web-Adresse in Deutschland, Österreich oder der Schweiz bestellt, kann die Berechnung der Anzahl Arbeitstage zu unterschiedlichen Ergebnissen führen.

Die Realität ist sogar noch etwas komplexer. Feiertage sind pro Bundesland, Kanton oder sogar Ortschaft unterschiedlich festgelegt. Der 6. Januar aus unserem Beispiel ist zwar in Österreich ein allgemeiner Feiertag, in Deutschland jedoch nur in den Bundesländern Baden-Württemberg, Bayern und Sachsen-Anhalt ein arbeitsfreier Tag. In der Schweiz ist der 6. Januar kein gesetzlicher Feiertag – außer in den Kantonen Uri, Schwyz und Tessin sowie in einzelnen Gemeinden in Graubünden. Dafür wird in der Schweiz am 2. Januar nicht gearbeitet – aber selbstverständlich auch nur in einzelnen Kantonen.

Solche Sachverhalte können in einer allgemeinen Zeitdimension nicht abgebildet werden. Deshalb kann es je nach Anforderungen notwendig sein, eine zusätzliche Zeitdimension für regionale Kalenderattribute (z.B. Feiertage und Ferien) einzuführen, die dann für entsprechende Abfragen verwendet werden kann.

In Abbildung 3 wird dies mit einem Beispiel illustriert. Die zusätzliche Dimensionstabelle DIM\_REGION\_DATE enthält für jede Kombination von Datum und Region (z.B. Bundesland) einen Datensatz, der über einen künstlichen Schlüssel REGION\_DATE\_ID referenziert wird und die regionalen Feiertage und Ferien definiert.

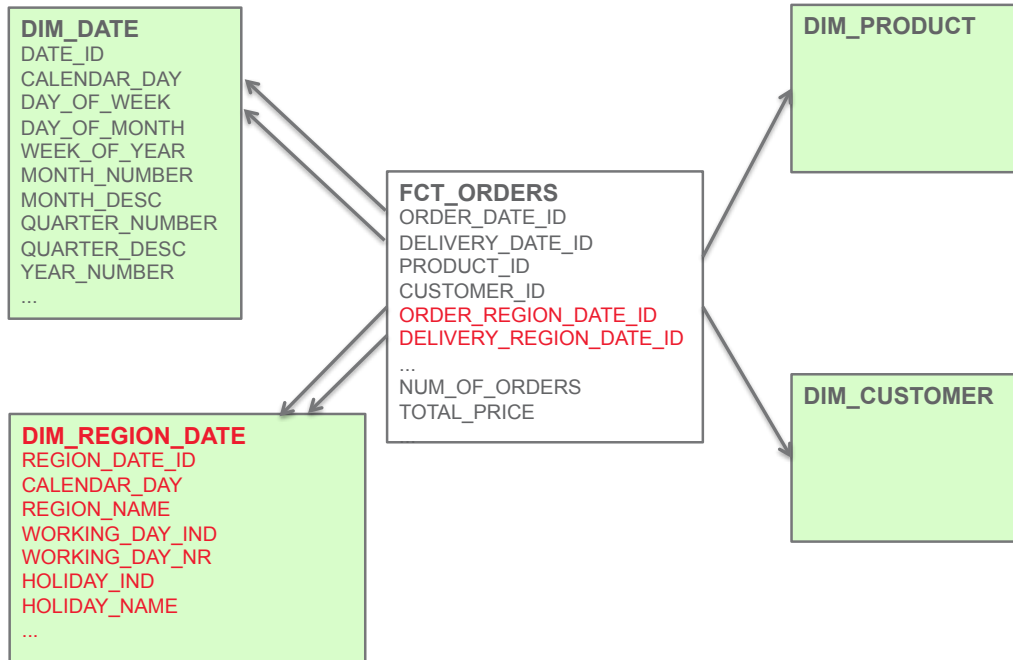


Abb. 3: Star Schema mit zusätzlicher regionaler Zeitdimension

Voraussetzung für Auswertungen auf diese regionale Zeitdimension ist es, dass die Fakten einer Region zugeordnet werden können – beispielsweise über welche Web-Adresse ein Kunde eine Bestellung aufgegeben hat. Diese Information wird als zusätzlicher Fremdschlüssel auf die regionale Zeitdimension in der Faktentabelle gespeichert und kann dann für Abfragen auf die regionale Zeitdimension verwendet werden:

```

SELECT od.calendar_day order_date
      , dd.calendar_day delivery_date
      , drd.working_day_num - ord.working_day_num AS num_of_working_days
FROM fct_orders f
JOIN dim_date od
  ON od.date_id = f.order_date_id
JOIN dim_date dd
  ON dd.date_id = f.delivery_date_id
JOIN dim_region_date ord
  ON ord.region_date_id = f.order_region_date_id
JOIN dim_region_date drd
  ON drd.region_date_id = f.delivery_region_date_id
  
```

Anhand dieses Beispiels wird ersichtlich, dass es zwar problemlos möglich ist, spezielle Anforderungen wie zeitlich unterschiedliche Kalenderdefinitionen in einem dimensionalen Datenmodell abzubilden, dass diese aber sowohl von den ETL-Prozessen beim Füllen der Faktentabellen sowie bei den Abfragen mittels SQL oder eines BI-Tools berücksichtigt werden müssen.

## Weitere Designtipps zur Zeitdimension

Die hier ausführlich beschriebene Problematik der korrekten Behandlung von Feiertagen ist nur eine – wenn auch häufig gestellte – Fragestellung im Zusammenhang mit dem Design von Zeitdimensionen. Daneben gibt es weitere interessante Aspekte, die zeigen, dass selbst eine auf den ersten Blick einfach erscheinende Anforderung – die Abbildung der Zeitachse im dimensionalen Modell – eine spannende Aufgabe sein kann. Weitere (einfachere) Designtipps zur Zeitdimension werden während des Vortrags vorgestellt. So wird erläutert, weshalb eine Trennung von Datum und Uhrzeit in separate Dimensionen zu empfehlen ist, weshalb als Primary Key der Zeitdimension ein sprechender Schlüssel verwendet werden soll, wie Abfragen für „relative“ Zeitangaben wie „Vorjahr“ oder „Aktueller Monat“ mittels zusätzlichen Indikatoren vereinfacht werden können und wie die Zeitdimension mit einem einzelnen SQL-Statement gefüllt werden kann.

Kontaktadresse:

Dani Schnider  
Trivadis AG  
Europa-Strasse 5  
CH-8152 Glattbrugg

Telefon: +41(0)44-808 70 20  
Fax: +41(0)44-808 70 21  
E-Mail: [dani.schnider@trivadis.com](mailto:dani.schnider@trivadis.com)  
Internet: [www.trivadis.com](http://www.trivadis.com)