

Haben Sie die Zeit im Griff? Designtipps zur Zeitdimension

DOAG Konferenz 2012

Dani Schnider
Trivadis AG

Nürnberg, 22. November 2012

BASEL BERN LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

1

2012 © Trivadis

DOAG - Designtipps zur Zeitdimension
22. November 2012

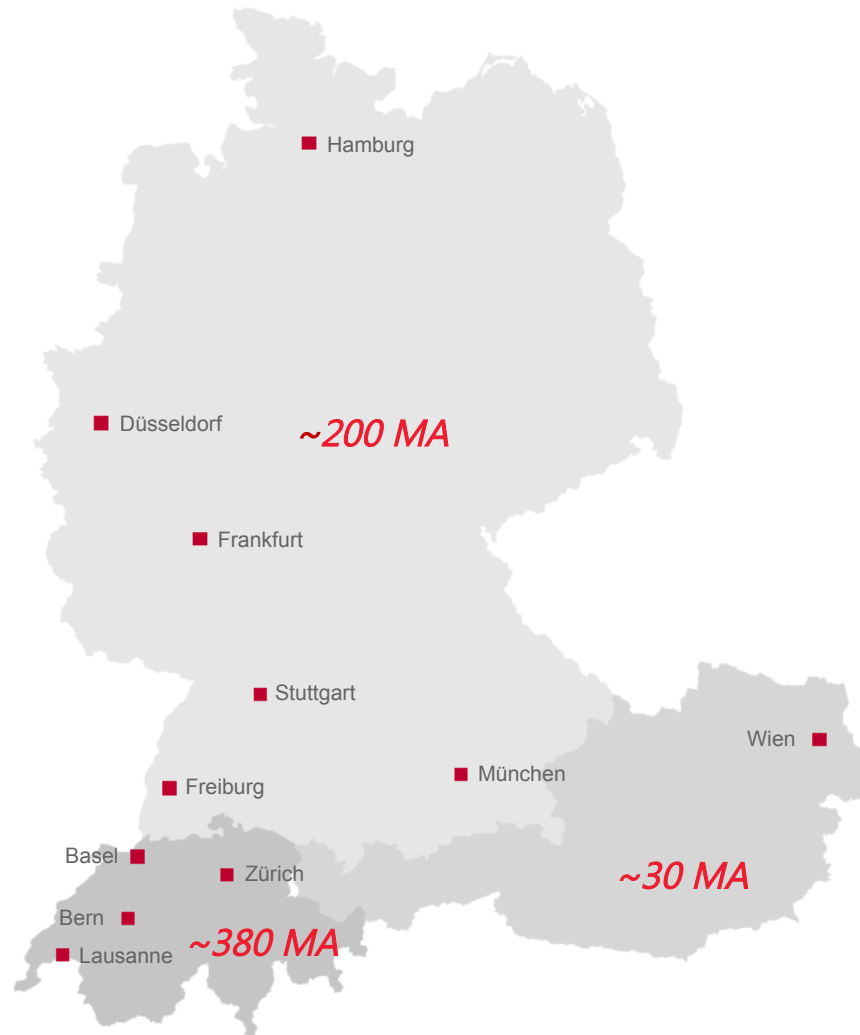
trivadis
makes IT easier. ■ ■ ■

Dani Schnider

- Principal Consultant und DWH/BI Lead Architect bei Trivadis in Zürich
- Kursleiter für Trivadis-Kurse über Data Warehousing, SQL Optimierung und Oracle Warehouse Builder
- Co-Autor des Buches «Data Warehousing mit Oracle»



Trivadis Facts & Figures



11 Trivadis Niederlassungen mit über 600 Mitarbeitenden

Finanziell unabhängig und nachhaltig profitabel

Kennzahlen 2011

- Umsatz CHF 104 / EUR 84 Mio.
- Dienstleistungen für über 800 Kunden in mehr als 1'900 Projekten
- 200 Service Level Agreements
- Mehr als 4'000 Trainingsteilnehmer
- Forschungs- und Entwicklungsbudget: CHF 5.0 / EUR 4 Mio.

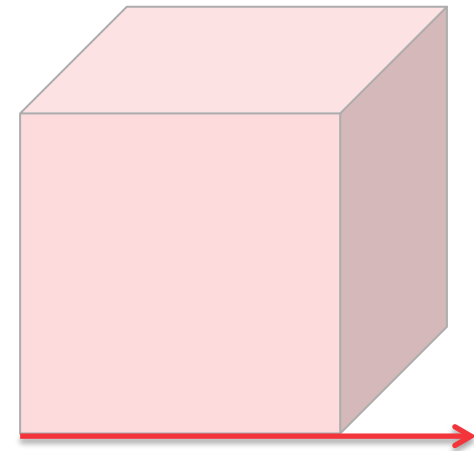
Die Zeit im Data Warehouse

Typische DWH-Abfragen:

- Mitarbeiterauslastung im letzten Quartal
- Umsatz pro Monat und Produktkategorie im laufenden Jahr
- Anzahl Transaktionen der letzten drei Tage
- Online-Bestellungen pro Wochentag und Land
- Vertragsabschlüsse pro Monat im laufenden Jahr im Vergleich zum Vorjahr

➔ Abfragen im DWH haben immer einen Zeitbezug

➔ Jeder Data Mart hat eine Zeitdimension

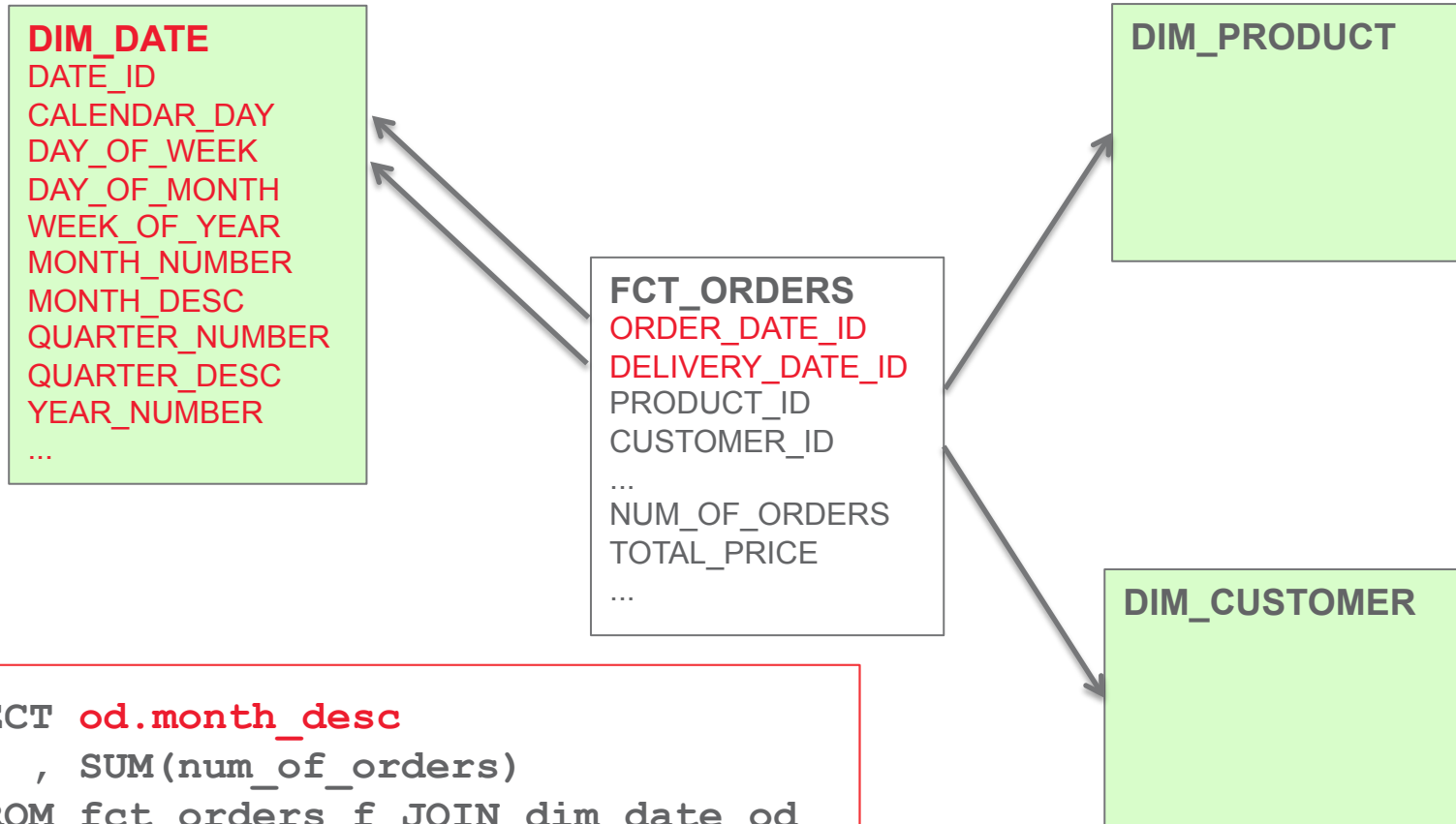




Zeitdimension

- **Allgemeine Designtipps**
- Umgang mit Feiertagen
- Füllen der Zeitdimension

Warum überhaupt eine explizite Zeitdimension?



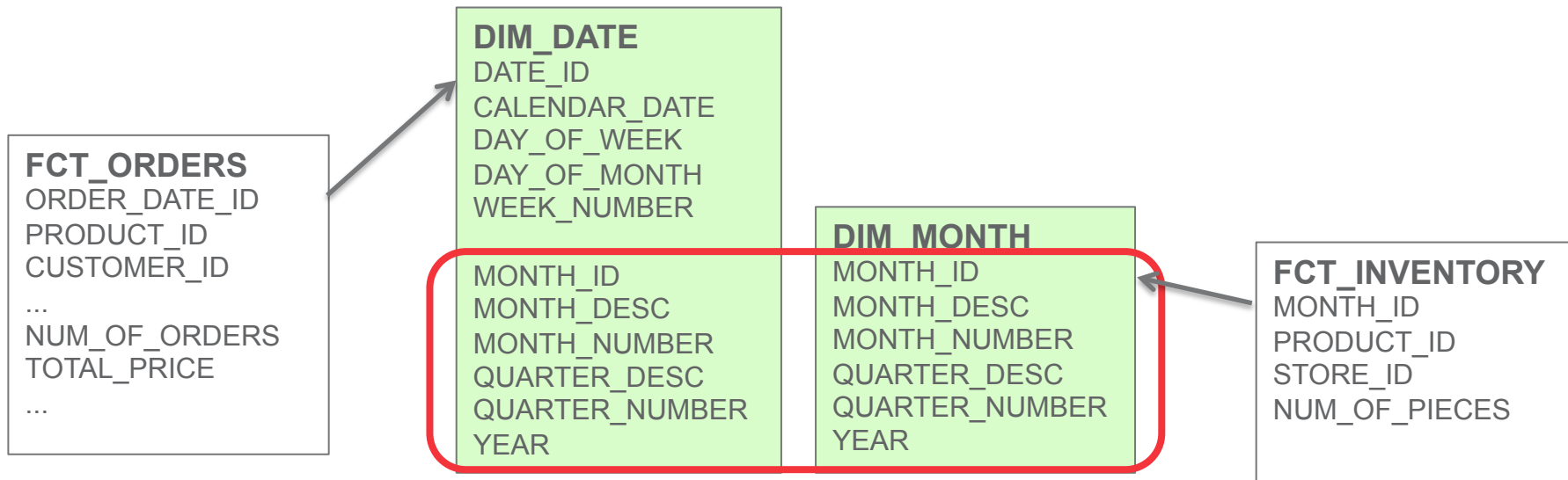
```
SELECT od.month_desc  
      , SUM(num_of_orders)  
FROM fct_orders f JOIN dim_date od  
  ON od.date_id = f.order_date_id  
GROUP BY od.month_desc  
WHERE od.quarter_desc = '2012/Q3'
```

Aufbau der Zeitdimension

CALENDAR_DAY	DAY_OF_WEEK	DAY_OF_MONTH	WEEK_OF_YEAR	MONTH_DESC	QUARTER_DESC	YEAR_NUMBER
15.11.2012	Donnerstag	15	46	November 2012	2012/Q4	2012
16.11.2012	Freitag	16	46	November 2012	2012/Q4	2012
17.11.2012	Samstag	17	46	November 2012	2012/Q4	2012
18.11.2012	Sonntag	18	46	November 2012	2012/Q4	2012
19.11.2012	Montag	19	47	November 2012	2012/Q4	2012
20.11.2012	Dienstag	20	47	November 2012	2012/Q4	2012
21.11.2012	Mittwoch	21	47	November 2012	2012/Q4	2012
22.11.2012	Donnerstag	22	47	November 2012	2012/Q4	2012
23.11.2012	Freitag	23	47	November 2012	2012/Q4	2012
24.11.2012	Samstag	24	47	November 2012	2012/Q4	2012
25.11.2012	Sonntag	25	47	November 2012	2012/Q4	2012
26.11.2012	Montag	26	48	November 2012	2012/Q4	2012
27.11.2012	Dienstag	27	48	November 2012	2012/Q4	2012
28.11.2012	Mittwoch	28	48	November 2012	2012/Q4	2012
29.11.2012	Donnerstag	29	48	November 2012	2012/Q4	2012
30.11.2012	Freitag	30	48	November 2012	2012/Q4	2012
01.12.2012	Samstag	1	48	Dezember 2012	2012/Q4	2012
02.12.2012	Sonntag	2	48	Dezember 2012	2012/Q4	2012
03.12.2012	Montag	3	49	Dezember 2012	2012/Q4	2012
04.12.2012	Dienstag	4	49	Dezember 2012	2012/Q4	2012
05.12.2012	Mittwoch	5	49	Dezember 2012	2012/Q4	2012

Granularität der Zeitdimension

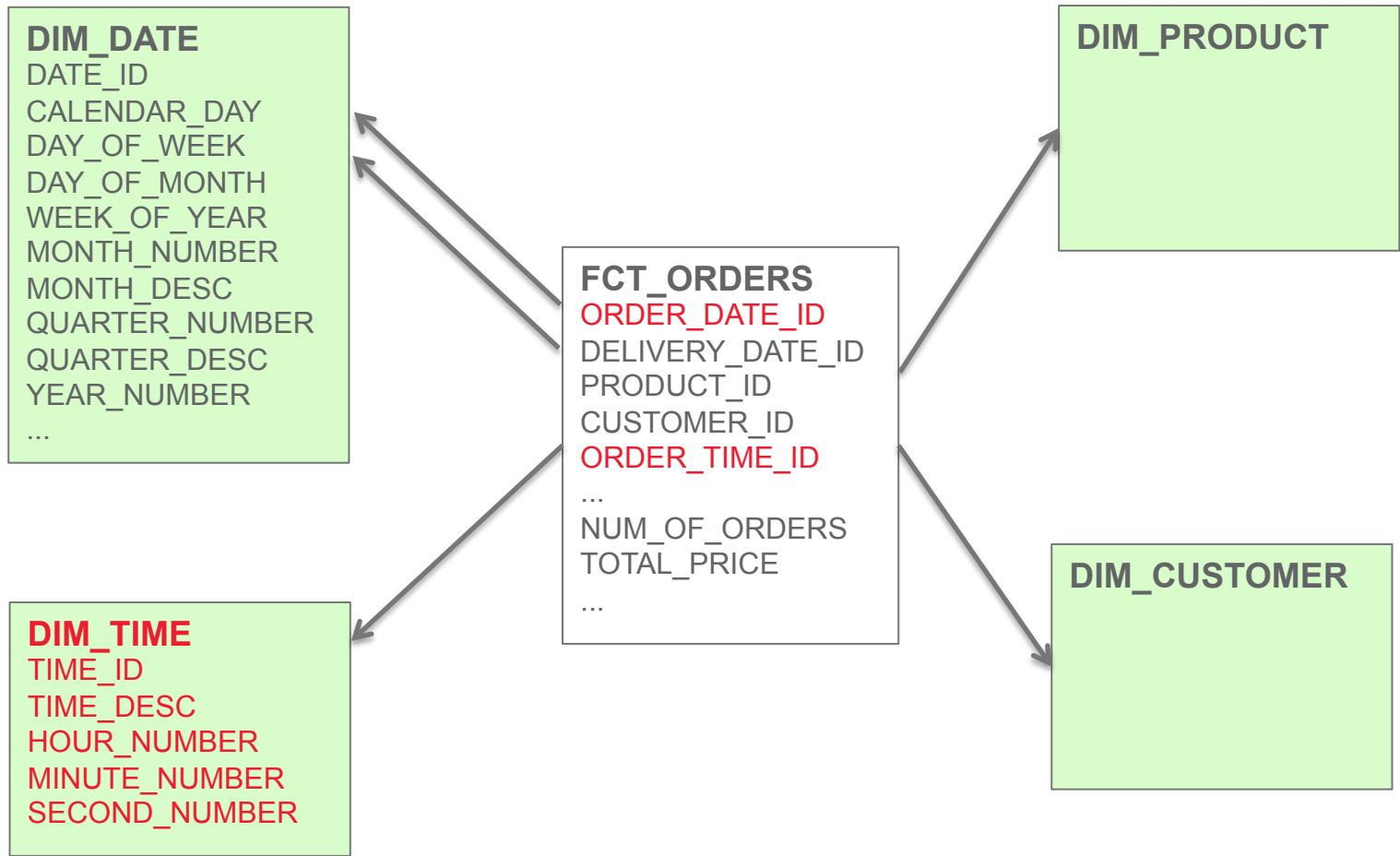
- Typische Granularität: Tag, Woche, Monat, Quartal, Jahr
- Bei unterschiedlichen Granularitäten: **Conformed Dimensions**



Anforderung: Bestellzeitpunkt mit Uhrzeit → **Granularität Sekunde?**

- 1 Tag hat 86400 Sekunden
- 1 Jahr hat 31536000 Sekunden

Designtipp 1: Datum und Uhrzeit trennen



Primary Key der Zeitdimension?

DIM_DATE

DATE_ID

CALENDAR_DAY

DAY_OF_WEEK

DAY_OF_MONTH

WEEK_OF_YEAR

MONTH_NUMBER

MONTH_DESC

QUARTER_NUMBER

QUARTER_DESC

YEAR_NUMBER

...

Datentyp und Inhalt von DATE_ID?

- Surrogate Key (Sequenznummer)
Datentyp NUMBER
- Datumwert
Datentyp DATE
- Datumwert im Format YYYYMMDD
Datentyp NUMBER

Zu beachten:

- Berechnung der Kardinalität durch Optimizer
- Zeitliche Partitionierung von Faktentabellen

Primary Key der Zeitdimension?

```
SELECT * FROM dim_date
WHERE date_id BETWEEN to_date('30.12.2011', 'dd.mm.yyyy')
AND to_date('09.01.2012', 'dd.mm.yyyy')
```

Id	Operation	Name	Rows
0	SELECT STATEMENT		12
* 1	TABLE ACCESS FULL	DIM_DATE	12

```
SELECT * FROM dim_date
WHERE date_id BETWEEN 20111230 AND 20120109
```

Id	Operation	Name	Rows
0	SELECT STATEMENT		348
* 1	TABLE ACCESS FULL	DIM_DATE	348

Designtipp 2: Primary Key der Zeitdimension

- Bei Granularität Tag: DATE_ID als Datum speichern
- Für alle anderen Stufen: „Sprechender“ Schlüssel vom Typ NUMBER

PK-Attribut	Granularität	Datentyp	Beispiel
DATE_ID	Tag	DATE	22-NOV-2012
MONTH_ID	Monat	NUMBER	201211
YEAR_ID	Jahr	NUMBER	2012
TIME_ID	Sekunde	NUMBER	152648

Designtipp 3: Abfrage-Indikatoren

DIM_DATE

DATE_ID
CALENDAR_DAY
DAY_OF_WEEK
DAY_OF_MONTH
WEEK_OF_YEAR
MONTH_NUMBER
MONTH_DESC
QUARTER_NUMBER
QUARTER_DESC
YEAR_NUMBER
CURR_WEEK_IND
LAST_WEEK_IND
NEXT_WEEK_IND
CURR_MONTH_IND
LAST_MONTH_IND
NEXT_MONTH_IND
CURR_MONTH_LAST_YEAR_IND
LAST_MONTH_LAST_YEAR_IND
...

- Indikatoren für "relative" Zeitangaben
 - Regelmässige Aktualisierung der Zeitdimension notwendig
- Beispiel:
 - Anzahl Bestellungen im aktuellen Monat im Vergleich zum Vorjahr

```
SELECT od.month_desc
       , SUM(num_of_orders)
FROM fct_orders f JOIN dim_date od
ON od.date_id = f.order_date_id
GROUP BY od.month_desc
WHERE od.curr_month_ind = 1
      OR od.curr_month_last_year_ind = 1
```

Anzahl Tage für Auslieferung einer Bestellung?



```
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)
```





Zeitdimension

- Allgemeine Designtipps
- **Umgang mit Feiertagen**
- Füllen der Zeitdimension

Designtipp 4: Umgang mit Arbeitstagen und Feiertagen

DIM_DATE

DATE_ID
CALENDAR_DAY
DAY_OF_WEEK
DAY_OF_MONTH
WEEK_OF_YEAR
MONTH_NUMBER
MONTH_DESC
QUARTER_NUMBER
QUARTER_DESC
YEAR_NUMBER
CURR_WEEK_IND
LAST_WEEK_IND
NEXT_WEEK_ID
CURR_MONTH_IND
LAST_MONTH_IND
NEXT_MONTH_ID
CURR_MONTH_LAST_YEAR_IND
LAST_MONTH_LAST_YEAR_IND
WORKING_DAY_IND
WORKING_DAY_NR
...

- WORKING_DAY_IND
 - 1 = Arbeitstag
 - 0 = Wochenende oder Feiertag
- WORKING_DAY_NR
 - Nummer des Arbeitstages seit Beginn der Zeitrechnung

Umgang mit Arbeitstagen und Feiertagen

CALENDAR_DAY	DAY_OF_WEEK	WORKING_DAY_IND	WORKING_DAY_NR
20.12.2011	Tuesday	1	1509
21.12.2011	Wednesday	1	1510
22.12.2011	Thursday	1	1511
23.12.2011	Friday	1	1512
24.12.2011	Saturday	0	1513
25.12.2011	Sunday	0	1513
26.12.2011	Monday	0	1513
27.12.2011	Tuesday	1	1513
28.12.2011	Wednesday	1	1514
29.12.2011	Thursday	1	1515
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

Anzahl Tage für Auslieferung einer Bestellung?



```
SELECT od.calendar_day order_date
      , dd.calendar_day delivery_date
      , dd.working_day_nr - od.working_day_nr 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)
```

Problem: Feiertage sind regional unterschiedlich





























Bayern



Kanton Zürich



Feiertag	Datum	Arbeitsfreie Kantone:																										
		 AG	 AI	 AR	 BE	 BL	 BS	 FR	 GE	 GL	 GR	 JU	 LU	 NE	 NW	 OW	 SG	 SH	 SO	 SZ	 TG	 TI	 UR	 VD	 VS	 ZG	 ZH	
Neujahrstag	1. Januar	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Berchtoldstag	2. Januar	a ¹			A			D		D	d	B	D	D ¹		D		D	D		A				A		D	D
Heilige Drei Könige	6. Januar																			B		A	B					
Josefstag	19. März											a		a		B				A		B	B			A	a	
Karfreitag	Freitag vor Ostern	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		A	A		A	A
Ostermontag	Montag nach Ostern	a	A	A	A	A	A	D	A	A	A	A	D	D	D	D	A	A	a	B	A	A	B	A		D	A	
Tag der Arbeit	1. Mai	D ²				A	A	c				A		A				B	C		B	B					A	
Auffahrt	39 Tage nach Ostern	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Pfingstmontag	50 Tage nach Ostern	a	A	A	A	A	A	D	A	B	A	A	D	D	D	D	A	A	a	B	A	B	B	A		D	A	
Fronleichnam	60 Tage nach Ostern	a	A			a		a			a	A	A	a	A	A			a	A		B	A		A	A		
Bundesfeier	1. August	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Mariä Himmelfahrt	15. August	a	B			a		a			a	B	A		A	A			a	A		A	A		A	A		
Allerheiligen	1. November	a	B					a		A	a	B	A		A	A	A		a	A		A	A		A	A		
Mariä Empfängnis	8. Dezember	a	B					a			a		B		A	A			a	B		B	A		A	A		
Weihnachtstag	25. Dezember	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Stephanstag	26. Dezember	a ¹	A ¹	A ¹	A	A	A	D		A	A		A	¹	D	D	A	A	a	B	A	A	B ¹			D	A	

Erläuterungen:

A Den Sonntagen gleichgestellter Feiertag gemäss Art. 20a Abs. 1 ArG im gesamten Kanton

a Den Sonntagen gleichgestellter oder sonstiger gesetzlich anerkannter Feiertag nur in einigen Gemeinden des Kantons. Bei einigen weiträumig begangenen Ereignissen ist jedoch eventuell auch in den übrigen Kantonsteilen mit Einschränkungen oder Feierlichkeiten zu rechnen

B Gesetzlich anerkannter öffentlicher **Ruhetag** im gesamten Kanton

C Gesetzlich anerkannter halber Feiertag (meist ab 12.00 Uhr) im gesamten Kanton

c Gesetzlich anerkannter halber Feiertag (meist ab 12.00 Uhr) nur für Mitarbeiter im **öffentlichen Dienst**, wird in der Regel aber von allen Arbeitsgruppen begangen

D Gesetzlich *nicht* anerkannter Feiertag, an dem in der Regel aber trotzdem im gesamten Kanton Arbeitsruhe ist

d Gesetzlich *nicht* anerkannter Feiertag, an dem in der Regel aber trotzdem in einigen bestimmten Gemeinden Arbeitsruhe ist

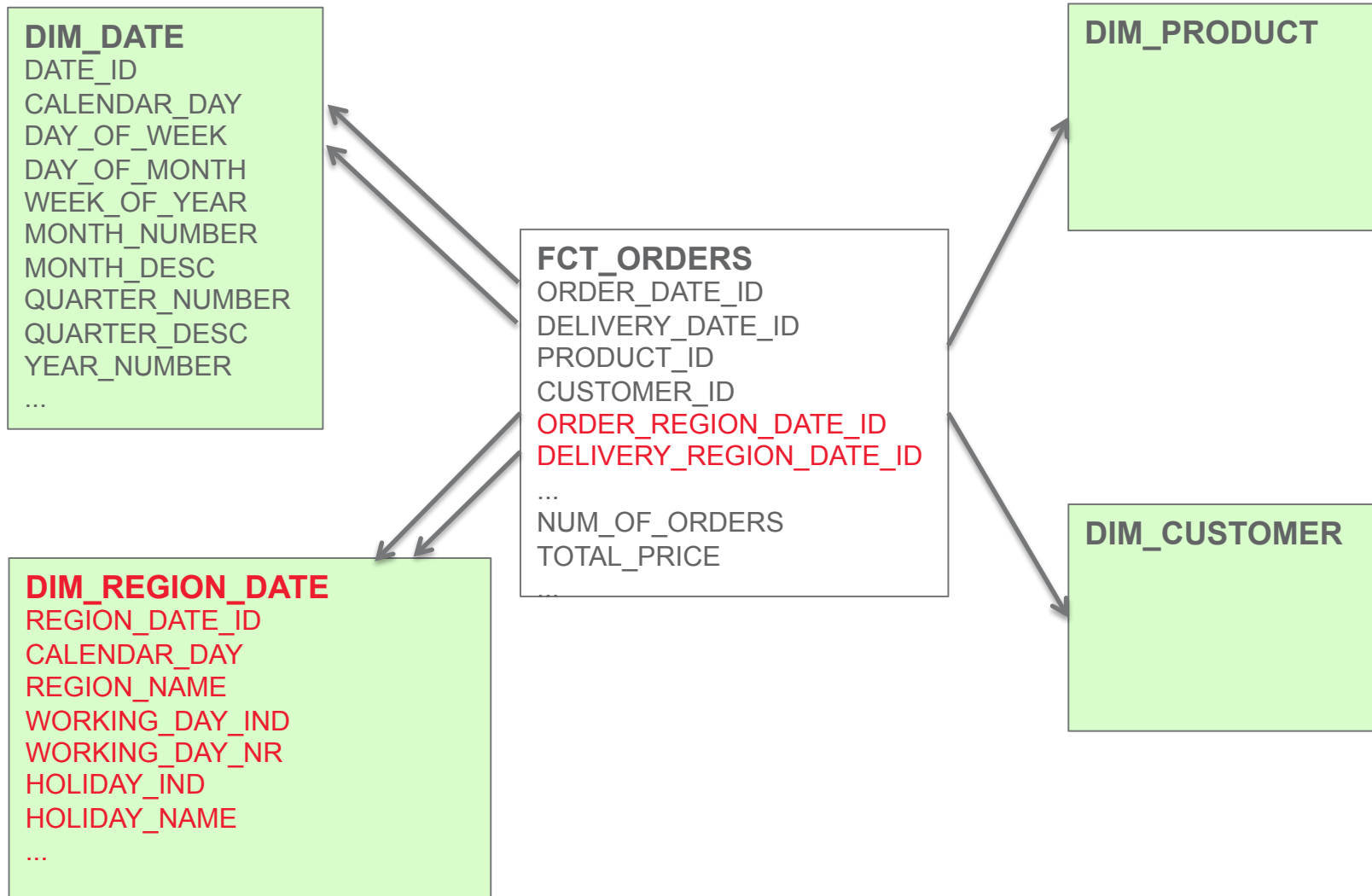
¹ Für den Berchtoldstag und den Stephanstag sind folgende Sonderfälle zu beachten:

- In den betreffenden Gemeinden von **AG** sind beide Tage nur dann gesetzlich anerkannt, wenn sie nicht auf einen **Dienstag** oder **Samstag** fallen; für den Stephanstag gilt dies ausserdem in den Kantonen **AI**, **AR** und **UR**. Der historische Grund für diese Regelung ist, dass es zwecks Sicherstellung der Grundversorgung niemals drei aufeinanderfolgende freie Tage geben sollte. Heute wird diese Vorschrift kaum noch beachtet, in der Regel ist also in jedem Jahr an beiden Tagen Arbeitsruhe.

20 • In **NE** sind beide Tage gesetzlich anerkannt, falls sie auf einen **Montag** fallen. In der Praxis wird allerdings üblicherweise der Berchtoldstag in jedem Jahr begangen, der Stephanstag hingegen nur, wenn er tatsächlich auf einen Montag fällt.

² Obwohl der 1. Mai im **AG** kein offizieller Feiertag ist, wird in der überwiegenden Mehrzahl der Betriebe nur bis mittags gearbeitet. Fällt der 1. Mai auf einen Montag, so ruht fast überall die Arbeit ganztägig.

Designtipp 5: Zusätzliche regionale Zeitdimension

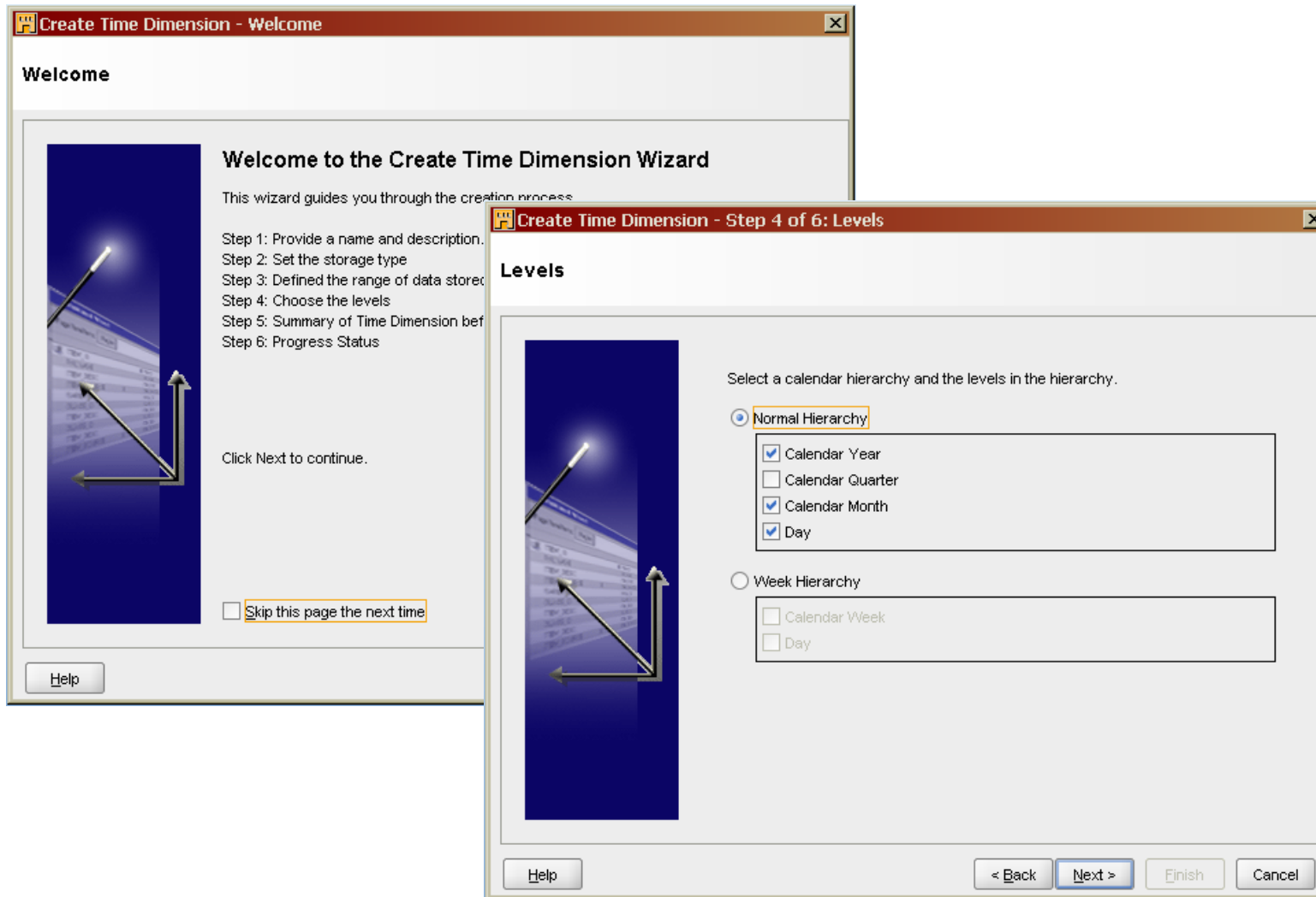




Zeitdimension

- Allgemeine Designtipps
- Umgang mit Feiertagen
- **Füllen der Zeitdimension**

OWB Time Dimension Wizard



PL/SQL-Prozedur

```
CREATE OR REPLACE PROCEDURE fill_dim_date (p_from_date IN DATE
                                           ,p_to_date   IN DATE)
IS
  v_dat DATE := p_from_date;
  v_row DIM_DATE%ROWTYPE;
BEGIN
  EXECUTE IMMEDIATE 'TRUNCATE TABLE dim_date';
  WHILE v_dat <= p_to_date LOOP
    v_row.date_id      := v_dat;
    v_row.calendar_day := v_dat;
    v_row.day_of_week  := TRIM(TO_CHAR(v_dat, 'Day'));
    v_row.day_of_month := TO_NUMBER(TO_CHAR(v_dat, 'DD'));
    v_row.week_of_year := TO_NUMBER(TO_CHAR(v_dat, 'IW'));
    v_row.month_number := TO_NUMBER(TO_CHAR(v_dat, 'MM'));
    ...
    INSERT INTO dim_date VALUES v_row;
    v_dat := v_dat + 1;
  END LOOP;
  COMMIT;
END fill_dim_date;
```


Designtipp 6: Füllen der Zeitdimension mit SQL-Statement

```
CREATE TABLE dim_date AS
WITH
  date_generator AS (
    SELECT TO_DATE('01.01.2006', 'DD.MM.YYYY') + ROWNUM
    - 1 AS calendar_day
    FROM dual
    CONNECT BY ROWNUM <= 10000
  )
, holidays AS (
  SELECT TO_DATE(tag, 'DD.MM.YYYY') AS calendar_day
  , feiertage AS text
  FROM feiertage_tmp
)
, working_days AS (
  SELECT d.calendar_day
  FROM date_generator d
  LEFT JOIN holidays h ON (d.calendar_day =
h.calendar_day)
  WHERE TO_CHAR(d.calendar_day, 'DY',
'NLS_DATE_LANGUAGE = english') NOT IN ('SAT', 'SUN')
  AND h.calendar_day IS NULL
)
, all_days AS (
  SELECT date_generator.calendar_day
  , TO_NUMBER(TO_CHAR
(date_generator.calendar_day, 'YYYY')) calendar_year
  , TO_NUMBER(TO_CHAR
(date_generator.calendar_day, 'YYYYMM')) calendar_month
  , TRIM(TO_CHAR(date_generator.calendar_day,
'Month')) month_name
  , CASE
    WHEN working_days.calendar_day IS NULL
  THEN 0
  ELSE 1
  END AS working_day_flag
  FROM date_generator
  LEFT JOIN working_days ON
working_days.calendar_day = date_generator.calendar_day
)
...
...
```

Lässt sich die Zeitdimension mit einem SQL-Statement füllen?



- DIM_DATE**
- DATE_ID
- CALENDAR_DAY
- DAY_OF_WEEK
- DAY_OF_MONTH
- WEEK_OF_YEAR
- MONTH_NUMBER
- MONTH_DESC
- QUARTER_NUMBER
- QUARTER_DESC
- YEAR_NUMBER
- ...



VIELEN DANK.

Trivadis AG

Dani Schnider

Europa-Strasse 5
CH-8152 Glattbrugg/Zürich
Schweiz

Tel. +41 44 808 70 20

Fax +41 44 808 70 21

info@trivadis.com

www.trivadis.com

BASEL BERN LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

26

2012 © Trivadis

DOAG - Designtipps zur Zeitdimension
22. November 2012

trivadis
makes IT easier. ■ ■ ■