

Een vaak voorkomende dimensie is de kalender, in elk Datawarehouse komt er wel een voor. Het is typisch zo'n dimensie die over het algemeen zo plat is als een dubbeltje. In dit voorbeeld maak ik gebruik van de volgende kalenderdimensie:

```
create table      kalender_dim (
id              number,
dag_nr         number,
dag_lang       varchar2(10),
maand_nr       number,
maand_lang     varchar2(10),
jaar_nr        number
);
```

Hierin zijn de dag en maand gegevens "platgeslagen". Uiteraard zijn de meeste kalendertabellen nog veel platter en uitgebreider. Maar ik beperk me even tot dag- en maandgegevens. Deze kalender_dim tabel is gevuld met de volgende inhoud:

```
select *
from kalender_dim
;
```

ID	DAG_NR	DAG_LANG	MAAND_NR	MAAND_LANG	JAAR_NR
1	20090102	FRIDAY	1	JANUARY	2009
2	20090103	SATURDAY	1	JANUARY	2009
3	20090104	SUNDAY	1	JANUARY	2009
4	20090105	MONDAY	1	JANUARY	2009
5	20090106	TUESDAY	1	JANUARY	2009
6	20090107	WEDNESDAY	1	JANUARY	2009
7	20090108	THURSDAY	1	JANUARY	2009
8	20090109	FRIDAY	1	JANUARY	2009
9	20090110	SATURDAY	1	JANUARY	2009
10	20090111	SUNDAY	1	JANUARY	2009
11	20090112	MONDAY	1	JANUARY	2009
12	20090113	TUESDAY	1	JANUARY	2009
13	20090114	WEDNESDAY	1	JANUARY	2009
14	20090115	THURSDAY	1	JANUARY	2009
... more					

Tevens een feitentabel waarin de verkopen per maand kunnen worden opgeslagen:

```
create table sales_facts_nok (  
    kalender_id      number,  
    sales            number  
);
```

Deze is bijvoorbeeld gevuld met de volgende data:

```
select *  
from sales_facts_nok;
```

KALENDER_ID	SALES
1	111
31	222
59	333
90	444
120	555
151	666
181	777
212	888
243	999
273	1010
304	1111
334	1212

Om vervolgens de verkopen per maand te zien, heeft de gebruiker de volgende query bij elkaar gesleept:

```
select      k.maand_nr,k.maand_lang, sum(f.sales)  
from        sales_facts_nok f, kalender_dim k  
where       f.kalender_id = k.id and  k.jaar_nr =2009  
group by    k.maand_nr,k.maand_lang  
order by    1;
```

De verkopen worden keurig weergegeven:

```

MAAND_NR MAAND_LANG SUM(F.SALES)
-----
1 JANUARY 111
2 FEBRUARY 222
3 MARCH 333
4 APRIL 444
5 MAY 555
6 JUNE 666
7 JULY 777
8 AUGUST 888
9 SEPTEMBER 999
10 OCTOBER 1010
11 NOVEMBER 1111
12 DECEMBER 1212

```

So far ... so good ... Nu is het ook mogelijk de kalender te normaliseren in een dag tabel en een maand tabel:

```

create table dag_dim (
id          number,
dag_nr      number,
dag_lang    varchar2(10),
maand_id    number,
);

create table maand_dim (
id          number,
maand_nr    number,
maand_lang  varchar2(10),
jaar_nr     number,
);

```

De maand tabel is gevuld met de volgende inhoud:

```

select * from maand_dim

ID MAAND_NR MAAND_LANG JAAR_ID
-----
1 1 JANUARY 2009
2 2 FEBRUARY 2009
3 3 MARCH 2009
4 4 APRIL 2009
5 5 MAY 2009
6 6 JUNE 2009
7 7 JULY 2009
8 8 AUGUST 2009
9 9 SEPTEMBER 2009
10 10 OCTOBER 2009
11 11 NOVEMBER 2009
12 12 DECEMBER 2009

```

Wederom een feitentabel:

```
create table sales_facts_ok (  
    maand_id    number,  
    sales       number  
);
```

Ook deze vullen we met verkoopgegevens:

```
select * from sales_facts_ok
```

MAAND_ID	SALES
1	111
2	222
3	333
4	444
5	555
6	666
7	777
8	888
9	999
10	1010
11	1111
12	1212

Om vervolgens de verkopen per maand te zien, heeft de gebruiker de volgende query bij elkaar gesleept:

```
select      m.maand_nr,m.maand_lang, sum(f.sales) from sales_facts_ok f, maand_dim m  
where      f.maand_id = m.id  
and        m.jaar_nr =2009  
group by   m.maand_nr,m.maand_lang  
order by 1;
```

Met als resultaat:

MAAND_NR	MAAND_LANG	SUM(F.SALES)
1	JANUARY	111
2	FEBRUARY	222
3	MARCH	333
4	APRIL	444
5	MAY	555
6	JUNE	666
7	JULY	777
8	AUGUST	888
9	SEPTEMBER	999
10	OCTOBER	1010
11	NOVEMBER	1111
12	DECEMBER	1212

Tot nu toe mag worden geconcludeerd dat de queries even eenvoudig zijn en het resultaat identiek. Maar nu komt het, de gebruiker sleept per ongeluk de dag uit de kalender dimensie en maakt de volgende query:

```
select      k.dag_lang, sum(f.sales)
from        sales_facts_nok f, kalender_dim k
where       f.kalender_id = k.id
and         k.jaar =2009
group      by m.dag_lang
```

En deze query geeft ook netjes de volgende (foutieve!) output:

DAG_LANG	SUM(F.SALES)
WEDNESDAY	1221
SATURDAY	888
SUNDAY	1666
THURSDAY	1010
FRIDAY	666
MONDAY	666
TUESDAY	2211

Uiteraard is dit hartstikke fout. Gelukkig is snel duidelijk dat hier een fout is begaan. Maar stelt u zich voor dat u te maken heeft met een complexe of zelfs meerdere dimensies? Is het dan nog wel zo duidelijk? Naast het feit dat een genormaliseerde dimensie contextueel beter is, voorkomt normaliseren ook dat dit soort fouten gemaakt kunnen worden. Tevens voorkomt normaliseren onnodige complexiteit, te lage granulariteit in feitentabellen, biedt het veel meer flexibiliteit en dwingt het tot nadenken over de informatiebehoefte van de klant.

Dus mijn advies: doe maar normaal(vorm).

Meer informatie over het genormaliseerde DWH is te vinden in Emiel zijn presentatie: "OBIEE the rising sun". Te downloaden op: www.bifacts.com/mydoc/obiee-the-rising-sun.pdf