

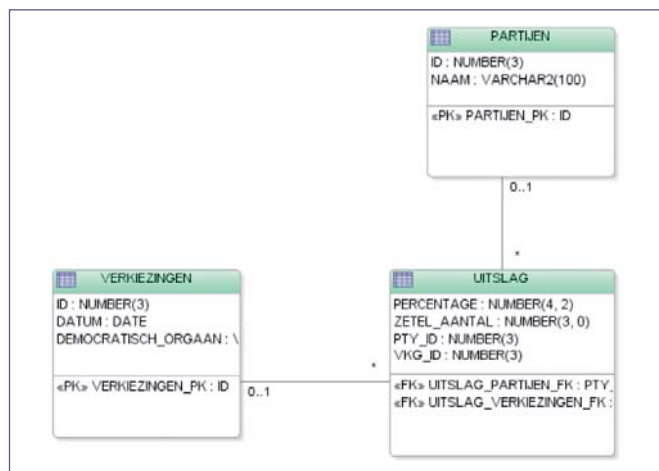
Politiek correct?

“Code schrijven is leuk”, hoorde ik laatst iemand zeggen. In eerste instantie was ik het daar helemaal mee eens. Toen ik er wat meer over na ging denken kwam ik tot de conclusie dat die stelling helemaal niet waar is. Het schrijven van code is niet leuk, het oplossen van een vraagstuk is leuk. Het puzzelen, het uitdokteren, het tot de meest efficiënte oplossing komen: dat is wat het werk zo leuk maakt.

De aardigheid van deze nieuwe rubriek in Optimize is dan ook voor een belangrijk deel dat puzzelen, de zoektocht naar de creatieve oplossing. Tegelijkertijd zullen in deze rubriek allerlei waardevolle, deels nieuwe en geavanceerde en deels al langer bestaande maar toch onbekende mogelijkheden van Oracle SQL de revue passeren. Er kan zoveel meer met SQL in Oracle dan we vaak denken. Het gebruik van Oracle 9i en 10g SQL-syntax en functies biedt mogelijkheden die eerder slechts bestonden door gebruik te maken van PL/SQL. Hopelijk inspireren de puzzels je die functionaliteit ook in de praktijk te gaan brengen!

De puzzel in Optimize wordt vergezeld van een online artikel. Op <http://technology.amis.nl/blog?p=1641> kunnen alle scripts worden gedownload, zowel de DDL-scripts om de tabellen te creëren en de data te laden als een SQL-script met mogelijke oplossingen voor de puzzels. Voor de eerste SQL-puzzel kijken we naar het kabinet Balkenende IV. Na de verkiezingen van 22 november 2006 werd er alom gemoepd dat het zo vreselijk moeilijk zou zijn om op basis van de uitslag een stabiel kabinet te vormen. Maar zijn alle mogelijkheden wel goed onderzocht? We gaan eens kijken aan de hand van enkele SQL query's wat nu precies de uitslag was op 22 november, wat zoal de mogelijkheden voor een driepartijkabinet waren en wat kleine verschuivingen in de verkiezingsuitslag aan nieuwe mogelijkheden zouden hebben opgeleverd.

Het datamodel dat we voor deze puzzel gebruiken staat



Dit datamodel bevat de uitslagen van de Tweede Kamer-verkiezingen van 2003 en 2006.

hieronder afgebeeld. In deze tabellen vind je de uitslagen van zowel de Tweede Kamer-verkiezingen van 2003 als die van 22 november 2006.

Presenteer verkiezingsuitslag

Het eerste vraagstuk: toon de uitslag van 22 november en geef aan welke winst of welk verlies in zetels iedere partij heeft gemaakt ten opzichte van de uitslag van 2003. Natuurlijk zijn er verschillende mogelijkheden om dit op te lossen. Wij hebben hier gekozen voor toepassing van Analytische Functies, geïntroduceerd in Oracle 8i. Ze bieden onder meer de mogelijkheid om data op te halen uit andere records zonder een self-join uit te voeren:

```
lead (zetel_aantal) over (partition by pty.naam
                        order by vkg.datum desc
                        )
```

Dit fragment bepaalt de waarde van zetel_aantal in de eerstvolgende rij voor dezelfde partij (partition by pty.naam),

met rijen gesorteerd op op verkiezingsdatum (order by vkg.datum desc) van nieuw naar oud. Deze expressie geeft ons de kern van het gevraagde resultaat:

NAAM	DATUM	ZETEL_AANTAL	LD
CDA	22-11-2006	41	44
CDA	20-1-2002	44	
Christenunie	22-11-2006	6	3
Christenunie	20-1-2002	3	
D'66	22-11-2006	3	6
D'66	20-1-2002	6	

De waarde in de laatste kolom is gelijk aan het zetel_aantal van de volgende rij. Voor het CDA staat er in de laatste kolom 44, het aantal behaalde zetels bij de vorige Tweede Kamer-verkiezingen. Voor de tweede regel is er geen volgende uitslag voor het CDA, de volgende rij hoort immers bij de ChristenUnie. Dat is een andere 'partition' dan het CDA en dus staat er in de laatste kolom NULL. Een eenvoudige rekensommetje geeft het verschil in zetels ten opzichte van de vorige keer weer:

```

zetel_aantal -
    lead (zetel_aantal,1,0) over (partition by pty.naam
                                order by vkg.datum desc
                                ) as zetel_overschuiving

```

De twee extra argumenten in de aanroep van lead – 1 en 0 – geven respectievelijk aan hoeveel rijen we verder willen kijken (één in dit geval) en welke waarde moet worden aangenomen als de gevraagde rij niet beschikbaar is. Door het tussentijdse resultaat in een subquery te plaatsen kunnen we in de buitenste query verder filteren tot we het beoogde resultaat hebben:

```

select *
from (
select vkg.democratisch_organ||' op '||vkg.datum titel
,      pty.naam partij
,      usg.zetel_aantal zetelaantal
,      usg.zetel_aantal -
        lead (zetel_aantal, 1,0) over ( partition by pty.
naam
                                order by vkg.
datum desc
                                ) zetel_overschui-
ving
,      vkg.datum
from verkiezingen vkg
,      partijen    pty
,      uitslag     usg
where vkg.id = usg.vkg_id
and   pty.id = usg.pty_id
and   vkg.democratisch_organ = 'Tweede Kamer'
)

```

```

where extract(year from datum) = 2006
order
by   zetelaantal desc
;
TITEL                                PARTIJ                                ZETELAANTAL ZETEL_
VERSCHUIVING DATUM
-----
Tweede Kamer op 22-NOV-06 CDA                                41
-3 22-11-2006
Tweede Kamer op 22-NOV-06 PVDA                                33
-9 22-11-2006
Tweede Kamer op 22-NOV-06 SP                                  25
16 22-11-2006
Tweede Kamer op 22-NOV-06 VVD                                  22
-6 22-11-2006
Tweede Kamer op 22-NOV-06 PVDV                                9
9 22-11-2006
...

```

Mogelijke coalities

De tweede uitdaging vraagt je om op basis van de uitslag van 22 november op zoek te gaan naar alle mogelijke coalities. Meer specifiek: alle coalities bestaande uit drie partijen (de aanname is: met twee lukt het niet en vier kan nooit stabiel worden) met tenminste 76 zetels.

Er zijn verschillende mogelijkheden om alle unieke combinaties van drie partijen af te gaan en het resultaat te filteren op de som van drie zetelaantallen. Een redelijk rechtlijnige en goed leesbare aanpak maakt gebruik van de With-clause voor het 'voor-definiëren' van een in-line view. Dit is een krachtige methode om een complex probleem in deelstappen op te lossen, binnen één SQL-query. Voor veel lezers wellicht wennen dat een SQL-query niet met SELECT hoeft te beginnen... Overigens biedt de With-clause definitie nog enkele voordelen: de in-line view kan meerdere malen gerefereerd worden elders in de query en in-line views kunnen op elkaar worden gebaseerd.

De query die ons de mogelijke coalities presenteert:

```

with fracties as -- alle 2e Kamer fracties met hun zetelaantallen
(
select pty.naam
,      usg.zetel_aantal
,      row_number () over (order by zetel_aantal) rn
from   verkiezingen vkg
,      partijen    pty
,      uitslag     usg
where  vkg.id = usg.vkg_id
and    pty.id = usg.pty_id
and    extract (year from vkg.datum) = 2006
order
by     zetel_aantal
)
select f1.naam||'/'||f2.naam||'/'||f3.naam coalitie

```



```

,      f1.zetel_aantal + f2.zetel_aantal + f3.zetel_aantal zetel_aantal
from  fracties f1
      join
      fracties f2
on    (f1.rn > f2.rn) -- join alleen met een andere, kleinere frac-
tie

      join
      fracties f3
on    (f2.rn > f3.rn) -- join alleen met een andere, kleinere frac-
tie
where f1.zetel_aantal + f2.zetel_aantal + f3.zetel_aantal > 75
order
by    zetel_aantal -- sorteert de coalities op hun zetel-aantal

```

CDA/PVDA/SGP	76
CDA/PVDA/PvdD	76
CDA/PVDA/D'66	77
PVDA/SP/VVD	80
CDA/PVDA/Christenunie	80
...	

In deze query wordt allereerst een in-line view opgezet die ons de nieuwe Tweede Kamer-fracties oplevert, met hun zetelaantallen. In een eenvoudige join worden alle fracties gekoppeld aan alle andere fracties die kleiner zijn – een simpele methode om combinaties niet dubbel te onderzoeken – en aangezien we naar driepartijencoalities zoeken voeren we nogmaals zo'n join uit. Vervolgens wordt gefilterd op de combinaties waarvoor geldt dat het gezamenlijke zetelaantal hoger is dan 75. Het resultaat van deze query begint met:

COALITIE	ZETEL_AANTAL
-----	-----

...en geeft maar liefst tien mogelijke combinaties voor rekenkundig verantwoorde coalities. Er viel dus voldoende te formeren.

De online versie van dit artikel (<http://technology.amis.nl/blog/?p=1641>) graaft nog wat dieper en bespreekt ondermeer hoe we coalities vinden waar maximaal één verliezer in zit en hoe kleine verschuivingen in de uitslag hadden kunnen doorwerken op de mogelijke coalities. Deze laatste uitdaging bevat ondermeer een in SQL ontwikkelde 'string tokenizer'. In de volgende aflevering van 'Puzzelen met SQL' gaan we het programma-aanbod op de Nederlandse televisie doorlichten en op grond van vooraf opgegeven voorkeuren van kijkers ideale televisieavonden samenstellen.

Lucas Jellema en Alex Nuijten zijn werkzaam bij AMIS.

socho_IT

software voor uitgeverijen



Socho_IT is gespecialiseerd in 'software voor uitgeverijen'. Met ons eigen systeem Zeno_2 bedienen we ruim zeventig uitgeverijen in Nederland en België. Omdat we onze dienstverlening momenteel succesvol uitbreiden zoeken we

Software-ontwikkelaars

met een (bijna) afgeronde hbo-opleiding. Als spil in onze serviceverlening vanuit Huizen of actief op uiteenlopende projecten. De combinatie van techniek en contact met verschillende klanten maakt iedere dag anders en doet een beroep op zowel je sociale als je technische vaardigheden.

Socho_IT groeit, groei met ons mee. Samen bepalen we de richting. De arbeidsvoorwaarden zijn prima, dus mail je cv naar info@socho.nl of bel Marco Clerx of Bob van der Horst voor meer informatie!

Niet gedetacheerd, maar werken aan een mooi product voor verschillende klanten? Dat kan bij Socho_IT!

voor meer informatie:
info@socho.nl
 t. 035 . 67 20 621
www.socho.nl

