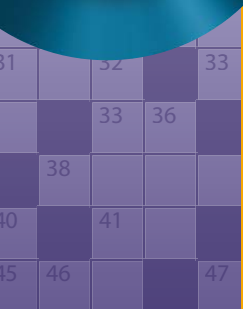


Puzzelen met SQL



Het is weer bijna december. Het paard van Sint Nicolaas staat bijna voor de deur. In sommige winkels zijn zelfs de eerste kerstkransjes te verkrijgen. Op televisie zijn de meeste reclames op de kinderen gericht. Het kan niet op met het speelgoed. En het is natuurlijk ook weer tijd om wat gezelschapsspellen uit de kast te halen, want je moet toch iets doen om die dagen te vullen.

Nu is het natuurlijk leuk om 'Mens Erger je Niet' te spelen, of als je met een grote groep bent 'Weerwolven van Wakkerdam' (een aanrader), maar deze spelen lenen zich niet echt voor SQL. Wat zich wel heel goed leent voor SQL zijn spelletjes zoals SET, maar hier hebben we in het verleden al een keertje een 'Puzzelen met SQL' aan gewijd. Deze keer gaan we MasterMind spelen. Het is een spel wat al lange tijd bestaat met eenvoudige regels.



MasterMind

Eerst maar even de regels toelichten, voordat we beginnen met puzzelen. Het doel van het spel is om een geheime combinatie van vier kleuren te ontdekken. Eén speler kiest uit de zes beschikbare kleuren er vier uit. Natuurlijk laat de speler deze niet aan de ander zien, anders is het wel heel gauw afgelopen. De andere speler moet de combinatie zien te raden, zowel de juiste kleur als de juiste positie van de kleuren. De eerste speler geeft terugkoppeling over de gekozen opstelling van de tweede speler. De terugkoppeling die gegeven wordt zegt iets over het correct zijn van kleur of kleur en positie. Maar welke kleur en welke positie, die informatie wordt niet gegeven.

De willekeurige keuze

Deze keer maken we geen gebruik van databasetabellen, maar doen we alles vanuit de query zelf. Voor zoiets eenvoudigs als MasterMind hebben we geen datamodel nodig.

Het spel gaat beginnen. Het eerste wat er moet gebeuren is dat de eerste speler een keuze dient te maken uit de beschikbare kleuren.

```
SQL> with kleuren as
  2 (select 'rood' kleur from dual union all
  3  select 'oranje'   from dual union all
  4  select 'zwart'   from dual union all
  5  select 'geel'    from dual union all
  6  select 'groen'   from dual union all
  7  select 'blauw'   from dual
  8 )
  9 select kleur, rownum lokatie
 10 from (select kleur
 11        from kleuren
 12         order by dbms_random.value)
 13 where rownum <= 4
 14 /
```

KLEUR	LOKATIE
-----	-----
groen	1
rood	2
oranje	3
zwart	4

Allereerst maken we gebruik van Subquery Factoring, de WITH clause in bovenstaande statement, om de kleuren te

definiëren. We hebben nu een tijdelijke view met de naam 'kleuren'. Uit deze kleuren maken we een willekeurige keuze door ze eerst te sorteren met behulp van DBMS_RANDOM en daarna te filteren waarbij we alleen de eerste vier rijen willen hebben. De inline view in regels 10 tot en met 12 is nodig om ervoor te zorgen dat eerst de sortering plaatsvindt, voordat de filtering gebeurt. Indien je geen gebruik maakt van een inline view, krijg je vier rijen die vervolgens met DBMS_RANDOM willekeurig gesorteerd wordt. En dat is niet de bedoeling. Laat je dus niet verleiden om dit te doen:

```
SQL> with kleuren as
 2 (select 'rood' kleur from dual union all
 3 select 'oranje' from dual union all
 4 select 'zwart' from dual union all
 5 select 'geel' from dual union all
 6 select 'groen' from dual union all
 7 select 'blauw' from dual
 8 )
 9 select kleur
10 , rownum lokatie
11 from kleuren
12 where rownum <= 4
13 order by dbms_random.value
14 /
```

KLEUR	LOKATIE
oranje	2
zwart	3
rood	1
geel	4

In bovenstaande query wordt er dus eerst gefilterd op vier rijen, die vervolgens worden gesorteerd.

Nu hebben we wel een keuze gemaakt maar eigenlijk willen we dit horizontaal geprojecteerd hebben in SQL, niet onder elkaar. Het uiteindelijke doel is om vier kolommen te krijgen, iedere kolom staat voor een lokatie, met daarin de gekozen kleur. Aangezien we gebruik maken van een Oracle 11g database, kunnen we gebruik maken van de PIVOT functie.

```
SQL> with kleuren as
 2 (select 'rood' kleur from dual union all
 3 select 'oranje' from dual union all
 4 select 'zwart' from dual union all
 5 select 'geel' from dual union all
 6 select 'groen' from dual union all
 7 select 'blauw' from dual
 8 )
 9 , puzzel as
10 (select kleur
11 , rownum lokatie
12 from (select kleur
13 from kleuren
14 order by dbms_random.value)
15 where rownum <= 4
16 )
17 select *
18 from puzzel
19 pivot (max (kleur) for lokatie in (1,2,3,4))
20 /
```

1	2	3	4
rood			
	geel	zwart	
			groen

Per lokatie krijgen we nu een kolom te zien met daarin de kleur. Dat is het effect van de PIVOT, een heel krachtige functie. Een ander -misschien iets minder leuk- effect is dat de kolomnamen overeen komen met de "name" in de FOR conditie van de PIVOT. In ons geval levert dat dus kolommen op met de namen '1', '2', '3' en '4'.

Om deze te kunnen querien, moeten we de kolomnamen tussen dubbele aanhalingstekens zetten. In onderstaande query is dit te zien. Het is ook mogelijk om een kolom alias op te nemen in de PIVOT clause.

Omdat we nog steeds vier rijen als resultaat hebben, maar er eigenlijk maar één willen maken we gebruik van de groepsfunctie MAX, iedere andere groepsfunctie was ook goed geweest om hetzelfde resultaat te verkrijgen.

```
SQL> with kleuren as
 2 (select 'rood' kleur from dual union all
 3 select 'oranje' from dual union all
 4 select 'zwart' from dual union all
 5 select 'geel' from dual union all
 6 select 'groen' from dual union all
 7 select 'blauw' from dual
 8 )
 9 , puzzel as
10 (select kleur
11 , rownum lokatie
12 from (select kleur
13 from kleuren
14 order by dbms_random.value)
15 where rownum <= 4
16 )
17 select max ("1") "1"
18 , max ("2") "2"
19 , max ("3") "3"
20 , max ("4") "4"
21 from puzzel
22 pivot (max (kleur) for lokatie in (1,2,3,4))
23 )
24 /
```

1	2	3	4
zwart	oranje	blauw	rood

Alle mogelijke oplossingen

De keuze is gemaakt, die laten we nu even voor wat het is. Iedere kleur kan in principe op iedere positie worden geplaatst. Aangezien er zes kleuren in het spel zitten en er vier kleuren gekozen moeten worden om een mogelijke oplossing te kun-

nen zijn, zijn er dus 360 mogelijke combinaties. Dit aantal kun je wiskundig berekenen door te stellen dat je vier keer uit alle kleuren moeten kiezen. De eerste keer kun je uit 6 kleuren kiezen, de tweede keer nog maar uit 5, daarna 4 en dan 3. Dus er zijn $6 * 5 * 4 * 3 = 360$ mogelijkheden. Het is ook mogelijk om een variant van MasterMind te spelen waarbij je dezelfde kleur meerdere keren in een geheime combinatie mag gebruiken, of zelfs kleuren helemaal weglaat - eigenlijk een NULL introduceren. We spelen hier de eenvoudige manier. Vier kleuren kiezen die uniek in een combinatie moet voorkomen. Om inzichtelijk te maken welke dit dan zijn, gebruiken we de volgende query.

```
SQL> with kleuren as
2 (select 'rood'   kleur from dual union all
3  select 'oranje' from dual union all
4  select 'zwart'  from dual union all
5  select 'geel'   from dual union all
6  select 'groen'  from dual union all
7  select 'blauw'  from dual
8 )
9 select k1.kleur
10        , k2.kleur
11        , k3.kleur
12        , k4.kleur
13  from kleuren k1
14  join kleuren k2
15  on k1.kleur <> k2.kleur
```

```
16  join kleuren k3
17  on k1.kleur <> k3.kleur
18  and k2.kleur <> k3.kleur
19  join kleuren k4
20  on k1.kleur <> k4.kleur
21  and k2.kleur <> k4.kleur
22  and k3.kleur <> k4.kleur
23  /
```

KLEUR	KLEUR	KLEUR	KLEUR
rood	oranje	zwart	geel
rood	oranje	zwart	groen
rood	oranje	zwart	blauw
rood	oranje	geel	zwart
rood	oranje	geel	groen
rood	oranje	geel	blauw
...			
blauw	groen	oranje	geel
blauw	groen	zwart	rood
blauw	groen	zwart	oranje
blauw	groen	zwart	geel
blauw	groen	geel	rood
blauw	groen	geel	oranje
blauw	groen	geel	zwart

360 rows selected.

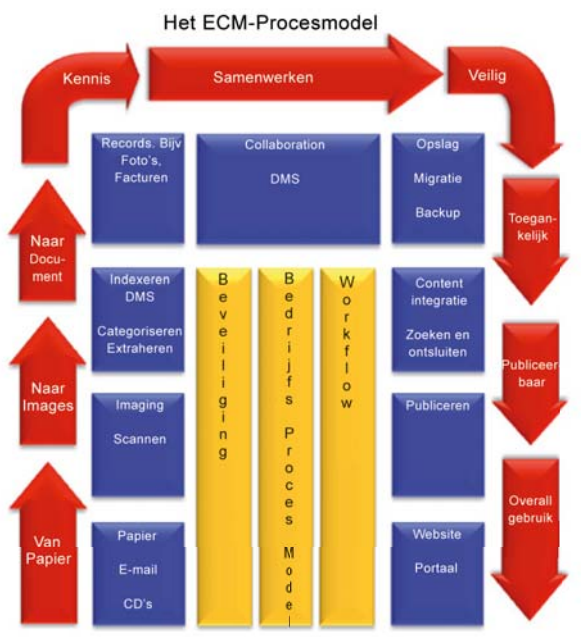
Om wat ruimte te sparen laten we hier niet alle 360 mogelijke combinaties zien. In bovenstaande query is de kracht van Subquery Factoring goed te zien. De "tabel" kleuren wordt in

(Advertentie)

ORAVISION

The Mid-Office Company

- OraVision, De integratie specialist**
- ✓ De bruggenbouwer tussen uw front - en backoffice
 - Waar Enterprise Application Integration en Enterprise Content Management elkaar ontmoeten
 - Waar primaire en documentprocessen naadloos samenwerken
- OraVision, De ECM Specialist**
- ✓ De specialist in het Oracle ECM platform
 - Oracle Collaboration Suite
 - Oracle Content Services
 - Oracle Content Database Suite
 - Oracle Enterprise Content Management Suite
 - Oracle Universal Content Management (voorheen Stellent)
 - ✓ Biedt producten en diensten voor het hele ECM - Procesmodel
 - ✓ Ons team van ervaren ICT - en ECM - specialisten maakt het verschil en is de sleutel tot uw succes!
 - ✓ Member of the BCT Group



Informatie ? www.oravision.nl

Of bel: 045 - 564 55 80

de query vier keer gebruikt, toch worden de individuele kleuren slechts éénmaal benoemd. Tevens maken we hier gebruik van de ANSI join syntax. Een veel gehoorde vraag is hoe om te gaan indien je meerdere tabellen aan elkaar wilt koppelen gebruik makend van de ANSI SQL join syntax. Gewoon doorknopen is het simpele antwoord. Omdat we afgesproken hebben om geen dubbele kleuren te gebruiken binnen de geheime combinatie filteren we deze eruit.

Men zegt dat het spelen van MasterMind het analytisch denken bevordert. Ook handig bij je dagelijkse werk.

De ontknoping

Nu zijn alle stukjes compleet. We hebben een manier om een geheime combinatie van vier kleuren te kiezen, en we hebben alle mogelijke oplossingen kunnen genereren. Nu wil alleen het geval dat je geen 360 kansen hebt om te bepalen welke combinatie er gekozen is, je zult het met twaalf kansen moeten doen. Bij iedere kans die je krijgt geeft de speler die de geheime combinatie weet je terugkoppeling over je keuze. Indien je een correcte kleur hebt, maar niet op de juiste plaats dan krijg je een zwart pinnetje. Als je een correcte kleur hebt die op de juiste plaats staat, dan krijg je een wit pinnetje. Natuurlijk wordt er dan niet bij gezegd welke kleur er dan juist is of welke kleur op de juiste plaats staat. Door deductie kun je uit verschillende kansen dan afleiden wat de correcte combinatie zou moeten zijn. De kans dat je het de eerste keer goed raad is erg klein. Men zegt dat het spelen van MasterMind je ook helpt om analytisch te denken, en dat kan je dus gebruiken in je dagelijks werk.

Maar we spelen natuurlijk niet helemaal eerlijk, we hebben tenslotte een Oracle Database tot onze beschikking. Om achter de geheime combinatie te komen maken we gebruik van de SET operator: INTERSECT.

```
SQL> with kleuren as
 2 (select 'rood'   kleur from dual union all
 3  select 'oranje' from dual union all
 4  select 'zwart'  from dual union all
 5  select 'geel'   from dual union all
 6  select 'groen'  from dual union all
 7  select 'blauw'  from dual
 8 )
 9 , puzzel as
10 (select kleur
11      , rownum lokatie
12      from (select kleur
13            from kleuren
```

```
14      order by dbms_random.value)
15  where rownum <= 4
16 )
17 select k1.kleur
18      , k2.kleur
19      , k3.kleur
20      , k4.kleur
21 from kleuren k1
22 join kleuren k2
23   on k1.kleur <> k2.kleur
24 join kleuren k3
25   on k1.kleur <> k3.kleur
26 and k2.kleur <> k3.kleur
27 join kleuren k4
28   on k1.kleur <> k4.kleur
29 and k2.kleur <> k4.kleur
30 and k3.kleur <> k4.kleur
31 intersect
32 select max ("1")
33      , max ("2")
34      , max ("3")
35      , max ("4")
36      from puzzel
37      pivot (max (kleur) for lokatie in (1,2,3,4));
```

```
KLEUR  KLEUR  KLEUR  KLEUR
-----
groen  rood   oranje  geel
```

En voila, daar is de 'geheime' combinatie.

De Set operator INTERSECT laat de overlappende records (in ons geval één record) tussen twee sets van data zien. De ene set data bestaat uit 360 mogelijke combinaties en de andere set uit het "geheime" record. Wat hebben die gemeenschappelijk? Alleen het record met de 'geheime' combinatie.

Zo, weer een puzzeltje opgelost, nu kunnen we dan toch 'Mens Erger je Niet' uit de kast halen, en onder het genot van een warme chocolade met een stukje boterletter een spelletje met de kinderen gaan doen. Fijne feestdagen en alvast de beste wensen voor volgend jaar. Veel puzzelplezier.

Referenties

- Om Mastermind online te spelen: <http://www.irt.org/games/js/mind/>
- Oracle documentatie over Set operatoren: http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/queries004.htm



Patrick Barel is consultant bij AMIS Services.

Hij schrijft op het blog van AMIS (<http://technology.amis.nl/blog>) en op zijn eigen blog (<http://blog.bar-solutions.com>).



Alex Nuijten is Oracle-consultant bij AMIS Services.

Hij schrijft op het blog van AMIS (<http://technology.amis.nl/blog>) en op zijn eigen blog (<http://nuijten.blogspot.com>).