



Op zoek naar aansluiting

Stel dat je een tabel hebt met begin- en einddatum, en dat sommige van de records een einddatum hebben die aansluitend is op een startdatum van een ander record. Om een overzicht te krijgen van aansluitende datum-ranges (bij gebrek aan een beter woord) kun je handig gebruik maken van analytische functies. Het uiteindelijke doel is om een overzicht te krijgen van de laagste startdatum tot en met de hoogste einddatum, waarbij er geen gaten in de datum-ranges mogen zitten.

Laten we beginnen met het tonen van de records zoals deze in de tabel staan.

```
create table date_ranges
as
select 1 id, to_date ('01-01-2007', 'dd-mm-yyyy') start_date, to_date
('31-01-2007', 'dd-mm-yyyy') end_date from dual union all
select 1 id, to_date ('01-02-2007', 'dd-mm-yyyy') start_date, to_date
('28-02-2007', 'dd-mm-yyyy') end_date from dual union all
select 1 id, to_date ('01-04-2007', 'dd-mm-yyyy') start_date, to_date
('30-04-2007', 'dd-mm-yyyy') end_date from dual union all
select 1 id, to_date ('01-05-2007', 'dd-mm-yyyy') start_date, to_date
('31-05-2007', 'dd-mm-yyyy') end_date from dual union all
select 1 id, to_date ('01-07-2007', 'dd-mm-yyyy') start_date, to_date
('31-08-2007', 'dd-mm-yyyy') end_date from dual
/
```

```
SQL> alter session set nls_date_format = 'dd-mm-yyyy'
2 /
```

Session altered.

```
SQL>
```

```
SQL> select *
2   from date_ranges
3 /
```

ID	START_DATE	END_DATE
1	01-01-2007	31-01-2007
1	01-02-2007	28-02-2007
1	01-04-2007	30-04-2007
1	01-05-2007	31-05-2007
1	01-07-2007	31-08-2007

Zoals je kunt zien zijn de eerste twee records aansluitende ranges. Deze twee records zouden als een record getoond moeten worden. Het uiteindelijke resultaat waar we naar op zoek zijn, is als volgt:

ID	START_DATE	END_DATE
1	01-01-2007	28-02-2007
1	01-04-2007	31-05-2007
1	01-07-2007	31-08-2007

Hoe zou je dit probleem nu kunnen aanpakken? Voor dit voorbeeld gaan we gebruikmaken van de analytische functie LAG. Met behulp van deze functie kunnen waarden uit records worden gelezen die al getoond worden. Deze functie haalt de waarde uit de rij die 'boven' de huidige rij staat.

```
SQL> select id
2   , start_date
3   , end_date
4   , lag (end_date) over (partition by id
5                          order by start_date
6                          ) lag_date
7*  from date_ranges
SQL> /
```

ID	START_DATE	END_DATE	LAG_DATE
1	01-01-2007	31-01-2007	
1	01-02-2007	28-02-2007	31-01-2007
1	01-04-2007	30-04-2007	28-02-2007
1	01-05-2007	31-05-2007	30-04-2007
1	01-07-2007	31-08-2007	31-05-2007

In het voorbeeld kun je zien dat de laatste kolom (LAG_DATE) de einddatum van het voorgaande record toont. Het verschil tussen de startdatum (START_DATE) en de LAG_DATE voor aansluitende ranges zou 1 dag moeten zijn. Als het niet aansluitend is, dan zal het verschil meer dan een dag bedragen. Aan de hand van dit gegeven kunnen we de rijen onderverdelen in groepjes.

```

SQL> select id
2      , start_date
3      , end_date
4      , lag (end_date) over (partition by id
5                                order by start_date
6                                ) lag_date
7      , case
8        when start_date
9          - lag (end_date) over (partition by id
10                                   order by start_date
11                                   )
12          = 1
13        then 0
14        else 1
15        end start_of_group
16 from date_ranges
17 /

```

ID	START_DATE	END_DATE	LAG_DATE	START_OF_GROUP
1	01-01-2007	31-01-2007		1
1	01-02-2007	28-02-2007	31-01-2007	0
1	01-04-2007	30-04-2007	28-02-2007	1
1	01-05-2007	31-05-2007	30-04-2007	0
1	01-07-2007	31-08-2007	31-05-2007	1

Nu weten we waar de verschillende 'snijpunten' moeten liggen, namelijk daar waar in de laatste kolom een '1' staat. Om er voor te zorgen dat de records die bij elkaar horen hetzelfde groepsgetal krijgen, maken we gebruik van de 'running total'-techniek die je met behulp van analytische functies kunt doen. De kolom waar we de running total op gaan doen, is de laatste kolom, de START_OF_GROUP. Voor de eerste twee kolommen is de running total 1, terwijl dit 2 is voor de derde en vierde kolom.

```

SQL> select id
2      , start_date
3      , end_date
4      , sum (start_of_group) over (partition by id
5                                order by start_date
6                                ) group_id
7 from (
8 select id
9      , start_date
10     , end_date
11     , lag (end_date) over (partition by id
12                               order by start_date
13                               ) lag_date
14     , case
15       when start_date
16         - lag (end_date) over (partition by id
17                                   order by start_date
18                                   )
19         = 1
20       then 0
21       else 1
22       end start_of_group
23 from date_ranges
24 )

```

```
25 /
```

ID	START_DATE	END_DATE	GROUP_ID
1	01-01-2007	31-01-2007	1
1	01-02-2007	28-02-2007	1
1	01-04-2007	30-04-2007	2
1	01-05-2007	31-05-2007	2
1	01-07-2007	31-08-2007	3

Nu is het een kwestie van de laagste startdatum en de hoogste einddatum per group_id en we zijn klaar.

```

SQL> select id
2      , min (start_date) start_date
3      , max (end_date) end_date
4 from (
5 select id
6      , start_date
7      , end_date
8      , sum (start_of_group) over (partition by id
9                                order by start_date
10                                   ) group_id
11 from (
12 select id
13      , start_date
14      , end_date
15      , lag (end_date) over (partition by id
16                               order by start_date
17                               ) lag_date
18      , case
19        when start_date
20          - lag (end_date) over (partition by id
21                                   order by start_date
22                                   )
23          = 1
24        then 0
25        else 1
26        end start_of_group
27 from date_ranges
28 ))
29 group by id, group_id
30 /

```

ID	START_DATE	END_DATE
1	01-01-2007	28-02-2007
1	01-04-2007	31-05-2007
1	01-07-2007	31-08-2007

In Oracle 10g is het ook mogelijk om het net iets anders te doen. Het '10g specifieke' aan deze query is het gebruik van IGNORE NULLS in regel 8. Het principe achter deze query is vrijwel gelijk aan de bovenstaande methode.

```

SQL> select id
2      , min (start_date)
3      , max (end_date)
4 from (

```

Adv

```

5 select id
6     , start_date
7     , end_date
8     , last_value (grp ignore nulls) over (partition by id
9                                             order by start_
date
10                                             ) g
11 from (
12 select id
13     , start_date
14     , end_date
15     , case
16         when (end_date + 1 =
17             lead (start_date
18                 ,1
19                 , end_date + 1) over (partition by id
20                                     order by start_
date

```

```

21                                     )
22                                     )
23         then rownum
24         end grp
25     from date_ranges
26 ))
27 group by id, g
28 /

ID MIN(START_ MAX(END_DA
-----
1 01-04-2007 31-05-2007
1 01-01-2007 28-02-2007
1 01-07-2007 31-08-2007

```

Alex Nuijten, AMIS Services BV

N I E U W S

Artikelen met praktische informatie, geschreven door en bestemd voor Oracle-professionals vindt u in het Online Archief van Array Publications. Vaktijdschriften als Database Magazine, Software Release en Java Magazine hebben hun artikelenarchief online gezet. Met een heldere zoekstructuur vindt u snel wat u zoekt op www.optimize.nl.

Samenwerking Oracle Think.com en Landelijke Stichting Tegen Zinloos Geweld

Oracle Think.com en de Landelijke Stichting Tegen Zinloos Geweld (LSTZG) gaan samenwerken om de Kids Tegen Geweld-leden van de LSTZG een online leercommunity te bieden. Aanstaaende vrijdag 6 juni wordt de samenwerking officieel afgetrapt met het plaatsen van de welbekende stoeptegels met het lieveheersbeestje bij het hoofdkantoor van Oracle in De Meern. Think.com van Oracle is een internationale leercommunity waarmee scholen, leraren en studenten over de hele wereld met elkaar samenwerken aan projecten en ervaringen uitwisselen om hun kennis te vergroten. Leraren kunnen eenvoudig werken met projecten en deze opnemen in hun lesprogramma. Kids Tegen Geweld is een project van de LSTZG en is speciaal opgericht voor kinderen tussen 8 en 14 jaar. De doelstelling van Kids Tegen Geweld is kinderen te laten opgroeien met het besef dat pesten, agressie en

geweld niet normaal zijn en ook nooit mogen worden. Kinderen denken zelf na over de veiligheid in hun eigen omgeving en bedenken zelf hoe de veiligheid vergroot kan worden in hun straat, wijk of bij hen op school. Zo kunnen ze op hun eigen manier een steentje bijdragen aan het verminderen van pesten, agressie en geweld. Kinderen kunnen zich zelfstandig aanmelden als Kids-clublid, of samen met andere kinderen een KidsTeam vormen. KidsTeams worden vooral gevormd door klassen op basisscholen. De LSTZG is al langere tijd op zoek naar een manier om hun 3500 kids-leden en 57 kidsteams met elkaar in contact te brengen. Zij willen meer interactie tussen de leden, zodat zij kunnen leren van elkaars acties en elkaar kunnen helpen bij pestproblemen. Think.com biedt de LSTZG nu een goede oplossing. Op Think.com zal er speciaal voor Kids Tegen Geweld een digitale omgeving worden gemaakt. Via deze omgeving kan de LSTZG haar leden opdrachten, tips en informatie verstrekken. Een voordeel van de digitale omgeving van Think.com is dat

het een veilige omgeving is. De toegang tot Think.com is beschermd met wachtwoorden en de inhoud wordt gecontroleerd door de leraren. De site bevat geen reclame, spam en virussen. De Oracle Education Foundation werkt voor Think.com samen met Symantec Corporation en biedt scholen in het basis- en middelbaar onderwijs daarmee een veilige en gratis webomgeving.

Informatie:

<http://www.oraclefoundation.org/>

Oracle neemt Admin Server over

Oracle blijft investeren in applicaties. Het bedrijf heeft een overeenkomst gesloten met betrekking tot de overname van Admin Server, leverancier van verzekeringspolisadministratiesoftware. Het zal deel gaan uitmaken van Oracle's bestaande software voor de verzekeringsbranche. De transactie zou in de eerste helft van dit jaar afgesloten moeten worden. Financiële details van de transactie zijn niet verstrekt.