



Barnett Newman, 1966-67, Who's afraid of red, yellow and blue III

Who's Afraid of Analytic Functions?

Alex Nuijten

**“Best thing since
SELECT statement”**

Tom Kyte

One Analytic Function can do more than a 1000 lines of code

Alex Nuijten



Analytic Functions *Revisited*

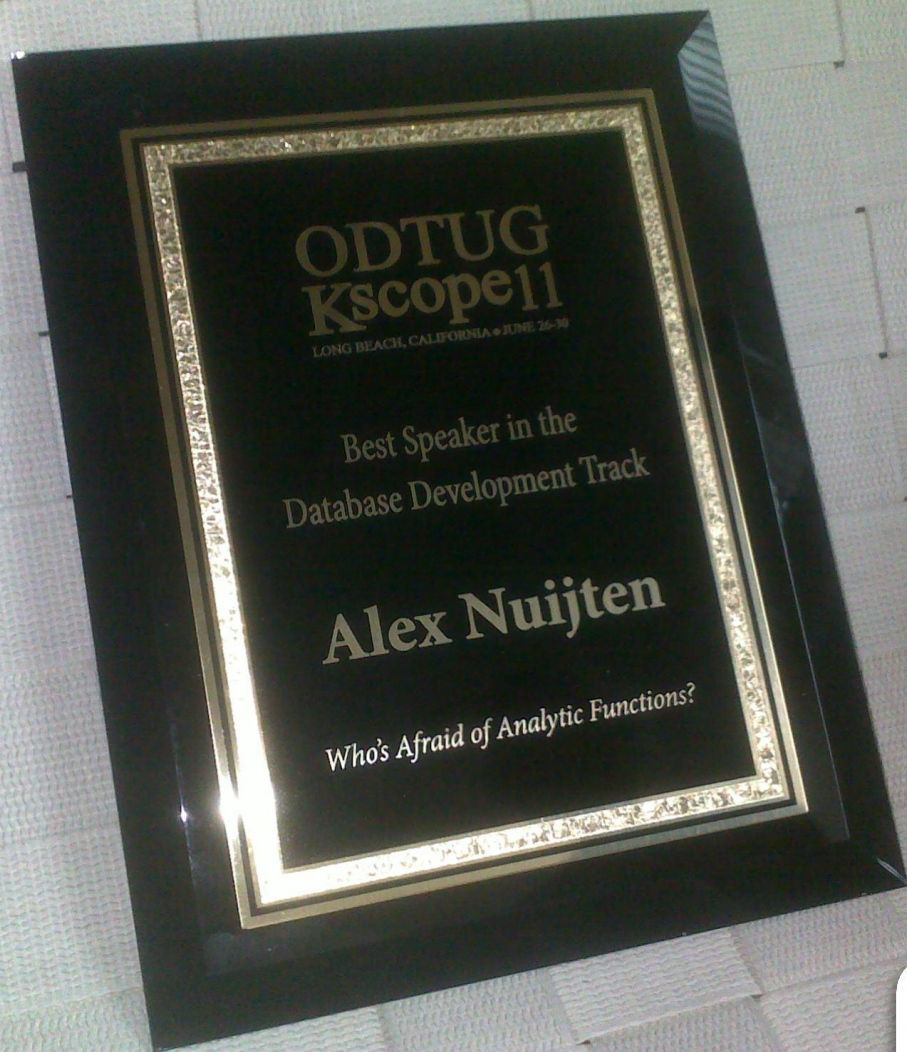
Alex Nuijten

AMIS

COMMITTED

UK OUG
UK ORACLE USER GROUP

ODTUG  **2010**
KALEIDOSCOPE
June 27 – July 1 • Marriott Wardman Park Hotel • Washington, D.C.





Oracle 8.1.6 EE

19

99

99









Who's Afraid of Analytic Functions?

- ✓ New
- ✓ Some Experience
- ✓ Advanced User

Result Set

Partition

Window

Result Set

Partition

Window

Result Set

Partition

Window

Result Set

Partition

Window

Result Set

Partition

Window

Result Set

Partition

Window

Result Set

Partition

Window

Direction of window movement

— Window Start —

Current Row: calculations based on window contents

— Window Finish —

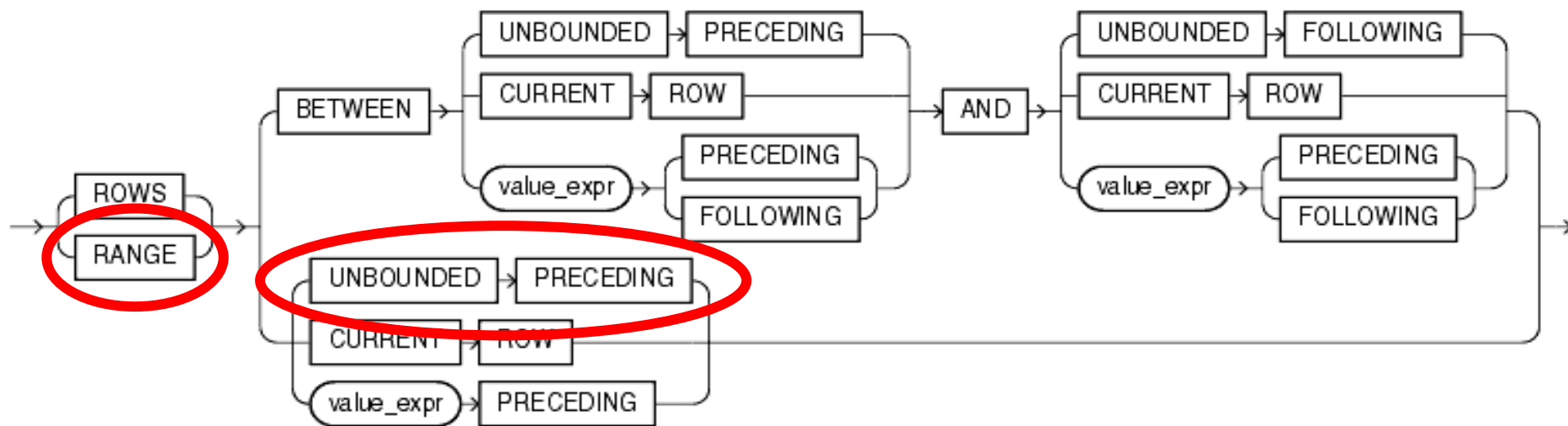
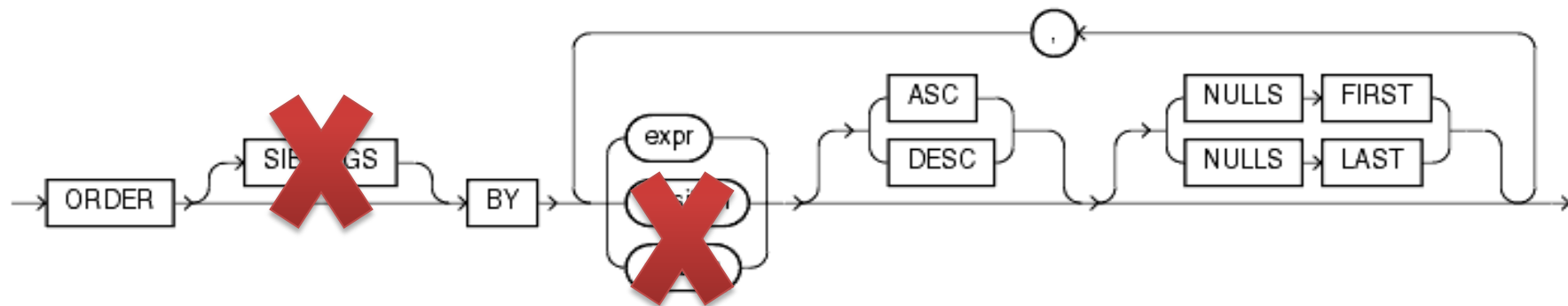
Size of the Window

- *Rows*: Number of rows

`rows between 2 preceding and 2 following`

- *Range*: Numeric offset

`range between 200 preceding and 250 following`



```
SQL> select ename, sal, sum (sal) over (order by ename) running_total
2      , sal
3      , sum (sal) over (order by ename) running_total
4      from emp
5      order by ename
6      /
```

Analytic Function

No Partition Clause

Default Window
Clause:
Range Unbounded
Preceding

ENAME	SAL	RUNNING_TOTAL
ADAMS	1100	
ALLEN	1600	
BLAKE	2850	5550
CLARK	2450	8000
FORD	3000	11000
JAMES	950	11950
JONES	2975	14925
KING	5000	19925
MARTIN	1250	21175
MILLER	1300	22475
SCOTT	3000	25475
SMITH	800	26275
TURNER	1500	27775
WARD	1250	29025

```

SQL> select ename
      2      , sal
      3      , sum (sal) over (order by ename) running_total
      4  from emp
      5  order by ename
      6  /

```

ENAME	SAL	RUNNING_TOTAL
-------	-----	---------------

ADAMS	1100	1100
ALLEN	1600	2700
BLAKE	2850	5550
CLARK	2450	8000
FORD	3000	11000
JAMES	950	11950
JONES	2975	14925
KING	5000	19925
MARTIN	1250	21175
MILLER	1300	22475
SCOTT	3000	25475
SMITH	800	26275
TURNER	1500	27775
WARD	1250	29025

```

SQL> select deptno
2         , ename
3         , sal
4         , sum (sal) over (partition by deptno
5                             order by ename
6                             ) dept_running_total
7   from emp
8  order by deptno
9         , ename;

```

DEPTNO	ENAME	SAL	DEPT_RUNNING_TOTAL
10	CLARK	2450	2450
10	KING	5000	7450
10	MILLER	1300	8750
20	ADAMS	1100	1100
20	FORD	3000	4100
20	JONES	2975	7075
20	SCOTT	3000	10075
20	SMITH	800	10875
30	ALLEN	1600	1600
30	BLAKE	2850	4450
30	JAMES	950	5400
30	MARTIN	1250	6650
30	TURNER	1500	8150
30	WARD	1250	9400

Filling in the



GAP

C	STARTDATE	VAL
-	- - - - -	- - - - -
A	24-SEP-10	QRS
A	24-OCT-10	
A	24-NOV-10	
A	24-DEC-10	a
A	24-JAN-11	
A	24-FEB-11	XY
A	24-MAR-11	ABC
A	24-APR-11	
A	24-MAY-11	
B	24-DEC-10	BLA
B	24-JAN-11	
B	24-FEB-11	
B	24-MAR-11	BLABLA
B	24-APR-11	

C	STARTDATE	VAL
-	- - - - -	- - - - -
A	24-SEP-10	QRS
A	24-OCT-10	↓
A	24-NOV-10	↓
A	24-DEC-10	a
A	24-JAN-11	↓
A	24-FEB-11	XY
A	24-MAR-11	ABC
A	24-APR-11	↓
A	24-MAY-11	↓
B	24-DEC-10	BLA
B	24-JAN-11	↓
B	24-FEB-11	↓
B	24-MAR-11	BLABLA
B	24-APR-11	↓

C	STARTDATE	VAL
-	- - - - -	- - - - -

A	24-SEP-10	QRS
---	-----------	-----

A	24-OCT-10	
---	-----------	--

A	24-NOV-10	
---	-----------	--

A	24-DEC-10	a
---	-----------	---

A	24-JAN-11	
---	-----------	--

A	24-FEB-11	XY
---	-----------	----

A	24-MAR-11	ABC
---	-----------	-----

A	24-APR-11	
---	-----------	--

A	24-MAY-11	
---	-----------	--

B	24-DEC-10	BLA
---	-----------	-----

B	24-JAN-11	
---	-----------	--

B	24-FEB-11	
---	-----------	--

B	24-MAR-11	BLABLA
---	-----------	--------

B	24-APR-11	
---	-----------	--


```
first_value (val)
  over (partition by code
        order by startdate
      ) fv
```

C	STARTDATE	VAL	FV
-	- - - - -	- - - - -	- - - - -
A	24-SEP-10	QRS	QRS
A	24-OCT-10		QRS
A	24-NOV-10		QRS
A	24-DEC-10	a	QRS
A	24-JAN-11		QRS
A	24-FEB-11	XY	QRS
A	24-MAR-11	ABC	QRS
A	24-APR-11		QRS
A	24-MAY-11		QRS
B	24-DEC-10	BLA	BLA
B	24-JAN-11		BLA
B	24-FEB-11		BLA
B	24-MAR-11	BLABLA	BLA
B	24-APR-11		BLA

```
select code
      , startdate
      , val
      , case
            when val is not null
            then 1
          end new_val
from tbl
;
```

C	STARTDATE	VAL	NEW_VAL
-	- - - - -	- - - - -	- - - - -
A	24-SEP-10	QRS	1
A	24-OCT-10		
A	24-NOV-10		
A	24-DEC-10	a	1
A	24-JAN-11		
A	24-FEB-11	XY	1
A	24-MAR-11	ABC	1
A	24-APR-11		
A	24-MAY-11		
B	24-DEC-10	BLA	1
B	24-JAN-11		
B	24-FEB-11		
B	24-MAR-11	BLABLA	1
B	24-APR-11		

```
select code
      , startdate
      , val
      , sum (new_val) over
          (partition by code
           order by startdate
          ) grp
from (select code
      , startdate
      , val
      , case
          when val is not null
          then 1
          end new_val
      from tbl
    ) ;
```

C	STARTDATE	VAL	GRP
---	-----------	-----	-----

-	- - - - -	- - - - -	- - - - -
---	-----------	-----------	-----------

A	24-SEP-10	QRS	1
---	-----------	-----	---

A	24-OCT-10		1
---	-----------	--	---

A	24-NOV-10		1
---	-----------	--	---

A	24-DEC-10	a	2
---	-----------	---	---

A	24-JAN-11		2
---	-----------	--	---

A	24-FEB-11	XY	3
---	-----------	----	---

A	24-MAR-11	ABC	4
---	-----------	-----	---

A	24-APR-11		4
---	-----------	--	---

A	24-MAY-11		4
---	-----------	--	---

B	24-DEC-10	BLA	1
---	-----------	-----	---

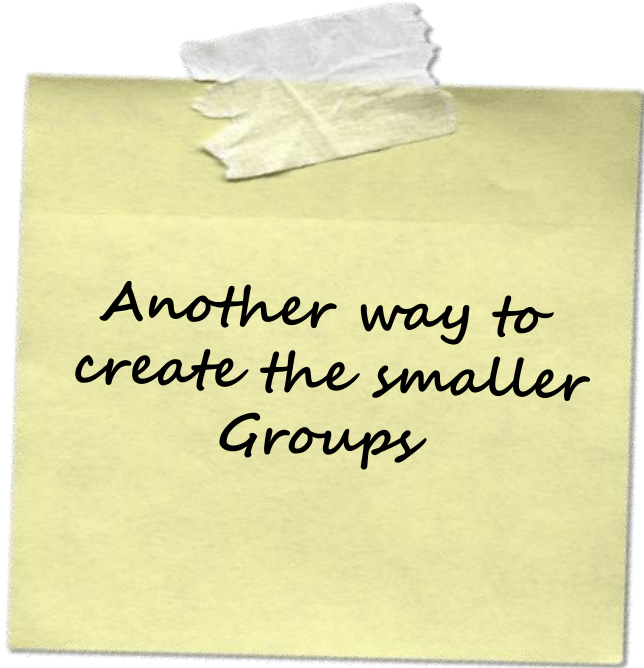
B	24-JAN-11		1
---	-----------	--	---

B	24-FEB-11		1
---	-----------	--	---

B	24-MAR-11	BLABLA	2
---	-----------	--------	---

B	24-APR-11		2
---	-----------	--	---

```
sum (nvl2 (val, 1,null))  
  over (partition by code  
        order by startdate  
      ) grp
```



Another way to
create the smaller
Groups

```

select code
      , startdate
      , val
      , first_value (val)
        over (partition by code
              , grp
              order by startdate
              ) new_val
from (select code
      , startdate
      , val
      , sum (new_val)
        over (partition by code
              order by startdate
              ) grp
      from (...)
    )
);

```


C	STARTDATE	VAL	NEW_VA
-	-----	-----	-----
A	24-SEP-10	QRS	QRS
A	24-OCT-10		QRS
A	24-NOV-10		QRS
A	24-DEC-10	a	a
A	24-JAN-11		a
A	24-FEB-11	XY	XY
A	24-MAR-11	ABC	ABC
A	24-APR-11		ABC
A	24-MAY-11		ABC
B	24-DEC-10	BLA	BLA
B	24-JAN-11		BLA
B	24-FEB-11		BLA
B	24-MAR-11	BLABLA	BLABLA
B	24-APR-11		BLABLA

Last_Value

```
last_value (val)
  over (partition by code
        order by startdate
      ) lv
```

C	STARTDATE	VAL	LV
-	- - - - -	- - - - -	- - - - -
A	24-SEP-10	QRS	QRS
A	24-OCT-10		
A	24-NOV-10		
A	24-DEC-10	a	a
A	24-JAN-11		
A	24-FEB-11	XY	XY
A	24-MAR-11	ABC	ABC
A	24-APR-11		
A	24-MAY-11		
B	24-DEC-10	BLA	BLA
B	24-JAN-11		
B	24-FEB-11		
B	24-MAR-11	BLABLA	BLABLA
B	24-APR-11		



```
select code
      , startdate
      , last_value (val ignore nulls)
                    over (partition by code
                        order by startdate
                        ) new_val
from tbl
```

C	STARTDATE	VAL	NEW_VA
-	-----	-----	-----
A	24-SEP-10	QRS	QRS
A	24-OCT-10		QRS
A	24-NOV-10		QRS
A	24-DEC-10	a	a
A	24-JAN-11		a
A	24-FEB-11	XY	XY
A	24-MAR-11	ABC	ABC
A	24-APR-11		ABC
A	24-MAY-11		ABC
B	24-DEC-10	BLA	BLA
B	24-JAN-11		BLA
B	24-FEB-11		BLA
B	24-MAR-11	BLABLA	BLABLA
B	24-APR-11		BLABLA

MIND THE GAP

```

SQL> select id
      2      , division
      3      , start_date
      4      , end_date
      5  from spantest
      6  where division = 1
      7      and start_date
      8          between to_date ('MAY 09 2011', 'MON DD YYYY')
      9          and to_date ('MAY 20 2011', 'MON DD YYYY')
     10 /

```

ID	DIVISION	START_DAT	END_DATE
-----	-----	-----	-----
1448	1	09-MAY-11	10-MAY-11
1449	1	11-MAY-11	12-MAY-11
1450	1	13-MAY-11	14-MAY-11
1452	1	17-MAY-11	18-MAY-11
1453	1	19-MAY-11	20-MAY-11

ID	DIVISION	START_DAT	END_DATE
1448	1	09-MAY-11	10-MAY-11
1449	1	11-MAY-11	12-MAY-11
1450	1	13-MAY-11	14-MAY-11
1452	1	17-MAY-11	18-MAY-11
1453	1	19-MAY-11	20-MAY-11



START_DAT	END_DATE
-----	-----
09-MAY-11	10-MAY-11
11-MAY-11	12-MAY-11
13-MAY-11	14-MAY-11
17-MAY-11	18-MAY-11
19-MAY-11	20-MAY-11



START_DAT	END_DATE
-----------	----------

-----	-----
-------	-------

- | | |
|-----------|-----------|
| 09-MAY-11 | 10-MAY-11 |
| 11-MAY-11 | 12-MAY-11 |
| 13-MAY-11 | 14-MAY-11 |
| 17-MAY-11 | 18-MAY-11 |
| 19-MAY-11 | 20-MAY-11 |



START_DAT	END_DATE
-----------	----------

-----	-----
-------	-------

- | | |
|-----------|-----------|
| 09-MAY-11 | 10-MAY-11 |
| 11-MAY-11 | 12-MAY-11 |
| 13-MAY-11 | 14-MAY-11 |
| 17-MAY-11 | 18-MAY-11 |
| 19-MAY-11 | 20-MAY-11 |

1

START_DAT	END_DATE
-----------	----------

-----	-----
-------	-------

- | | |
|-----------|-----------|
| 09-MAY-11 | 10-MAY-11 |
| 11-MAY-11 | 12-MAY-11 |
| 13-MAY-11 | 14-MAY-11 |
| 17-MAY-11 | 18-MAY-11 |
| 19-MAY-11 | 20-MAY-11 |

1

START_DAT	END_DATE
-----------	----------

-----	-----
-------	-------

- | | |
|-----------|-----------|
| 09-MAY-11 | 10-MAY-11 |
| 11-MAY-11 | 12-MAY-11 |
| 13-MAY-11 | 14-MAY-11 |
- | | |
|-----------|-----------|
| 17-MAY-11 | 18-MAY-11 |
| 19-MAY-11 | 20-MAY-11 |

>1

START_DAT	END_DATE
-----------	----------

-----	-----
-------	-------

- | | |
|-----------|-----------|
| 09-MAY-11 | 10-MAY-11 |
| 11-MAY-11 | 12-MAY-11 |
| 13-MAY-11 | 14-MAY-11 |
| 17-MAY-11 | 18-MAY-11 |
| 19-MAY-11 | 20-MAY-11 |

1

START_DAT	END_DATE
-----------	----------

-----	-----
-------	-------

- | | |
|-----------|-----------|
| 09-MAY-11 | 10-MAY-11 |
| 11-MAY-11 | 12-MAY-11 |
| 13-MAY-11 | 14-MAY-11 |
- | | |
|-----------|-----------|
| 17-MAY-11 | 18-MAY-11 |
| 19-MAY-11 | 20-MAY-11 |

Lag

```

SQL> select start_date
2         , end_date
3         , lag (end_date) over
4           (partition by division
5             order by start_date
6             ) lg_date
7   from spantest
8  where ...

```

START_DAT	END_DATE	LG_DATE
-----------	----------	---------

09-MAY-11	10-MAY-11	
11-MAY-11	12-MAY-11	10-MAY-11
13-MAY-11	14-MAY-11	12-MAY-11
17-MAY-11	18-MAY-11	14-MAY-11
19-MAY-11	20-MAY-11	18-MAY-11

```

SQL> select start_date
2         , end_date
3         , lag (end_date) over
4             (partition by division
5              order by start_date
6              ) lg_date
7   from spantest
8   where ...

```

START_DAT	END_DATE	LG_DATE
09-MAY-11	10-MAY-11	
11-MAY-11	12-MAY-11	10-MAY-11
13-MAY-11	14-MAY-11	12-MAY-11
17-MAY-11	18-MAY-11	14-MAY-11
19-MAY-11	20-MAY-11	18-MAY-11

```

SQL> select start_date
      2      , end_date
      3      , case
      4          when start_date -
      5              lag (end_date) over
      6                  (partition by division
      7                     order by start_date
      8                      ) = 1
      9              then 0
     10              else 1
     11          end span_group
     12 from spantest
     13 where ...

```

START_DAT	END_DATE	SPAN_GROUP
-----	-----	-----
09-MAY-11	10-MAY-11	1
11-MAY-11	12-MAY-11	0
13-MAY-11	14-MAY-11	0
17-MAY-11	18-MAY-11	1
19-MAY-11	20-MAY-11	0

```

SQL> select start_date
2         , end_date
3         , sum (span_group) over
4             (partition by division
5              order by start_date
6              ) grp
7   from ( ...)

```

START_DAT	END_DATE	GRP
-----	-----	-----
09-MAY-11	10-MAY-11	1
11-MAY-11	12-MAY-11	1
13-MAY-11	14-MAY-11	1
17-MAY-11	18-MAY-11	2
19-MAY-11	20-MAY-11	2

```

SQL> select min (start_date)
      2           , max (end_date)
      3         from (
      4       select
      5     ...
      6     28   ))
      7     29 group by grp
      8     30 /

```

MIN (START	MAX (END_D
-----	-----
09-MAY-11	14-MAY-11
17-MAY-11	20-MAY-11

旅人さん

Kenji Yamagishi

Aketi Jyuuzou

Tabi・Bito・San

Kenji Yamagishi

Aketi Jyuuzou

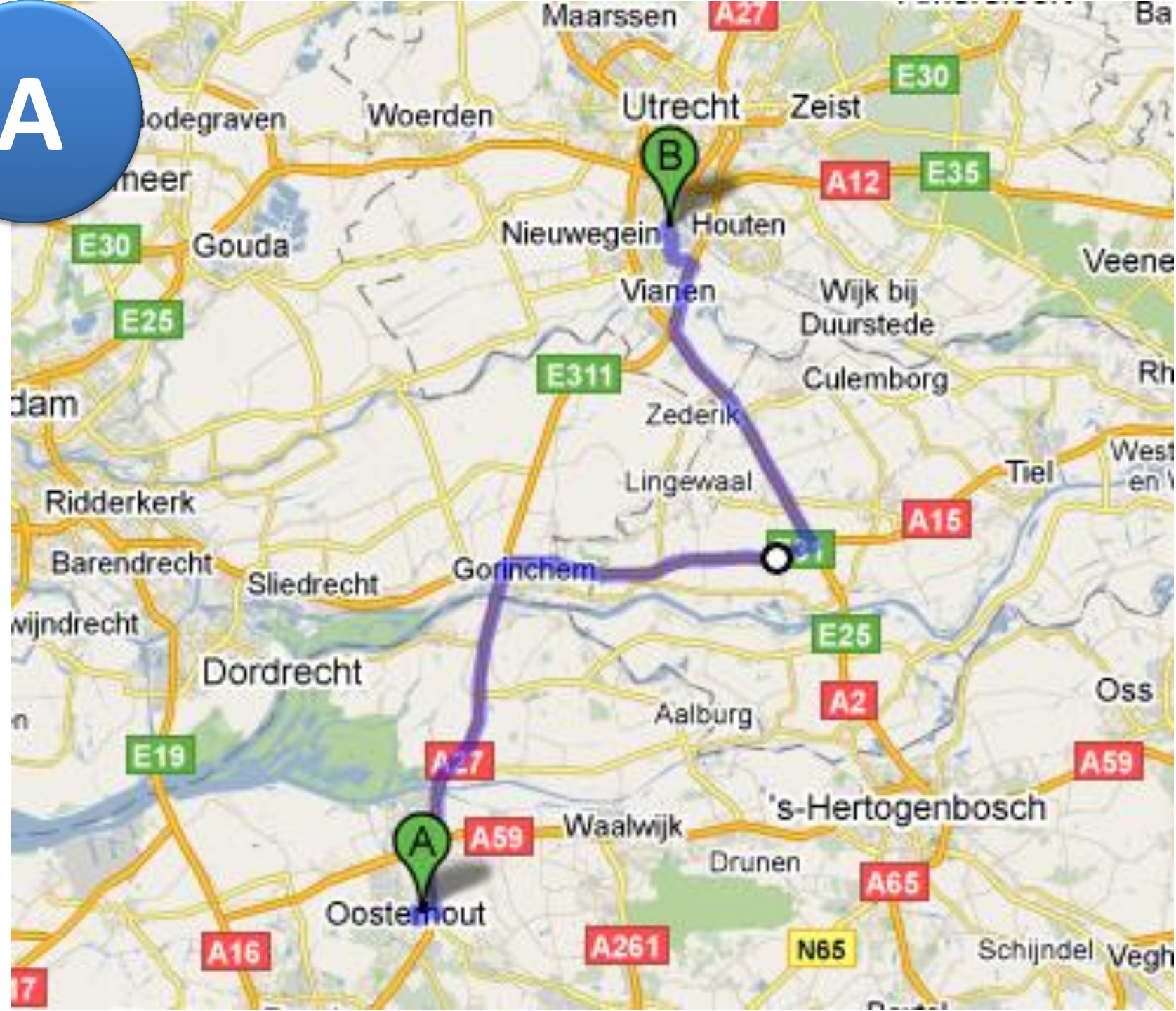
Honorable Traveller

Kenji Yamagishi

Aketi Jyuuzou

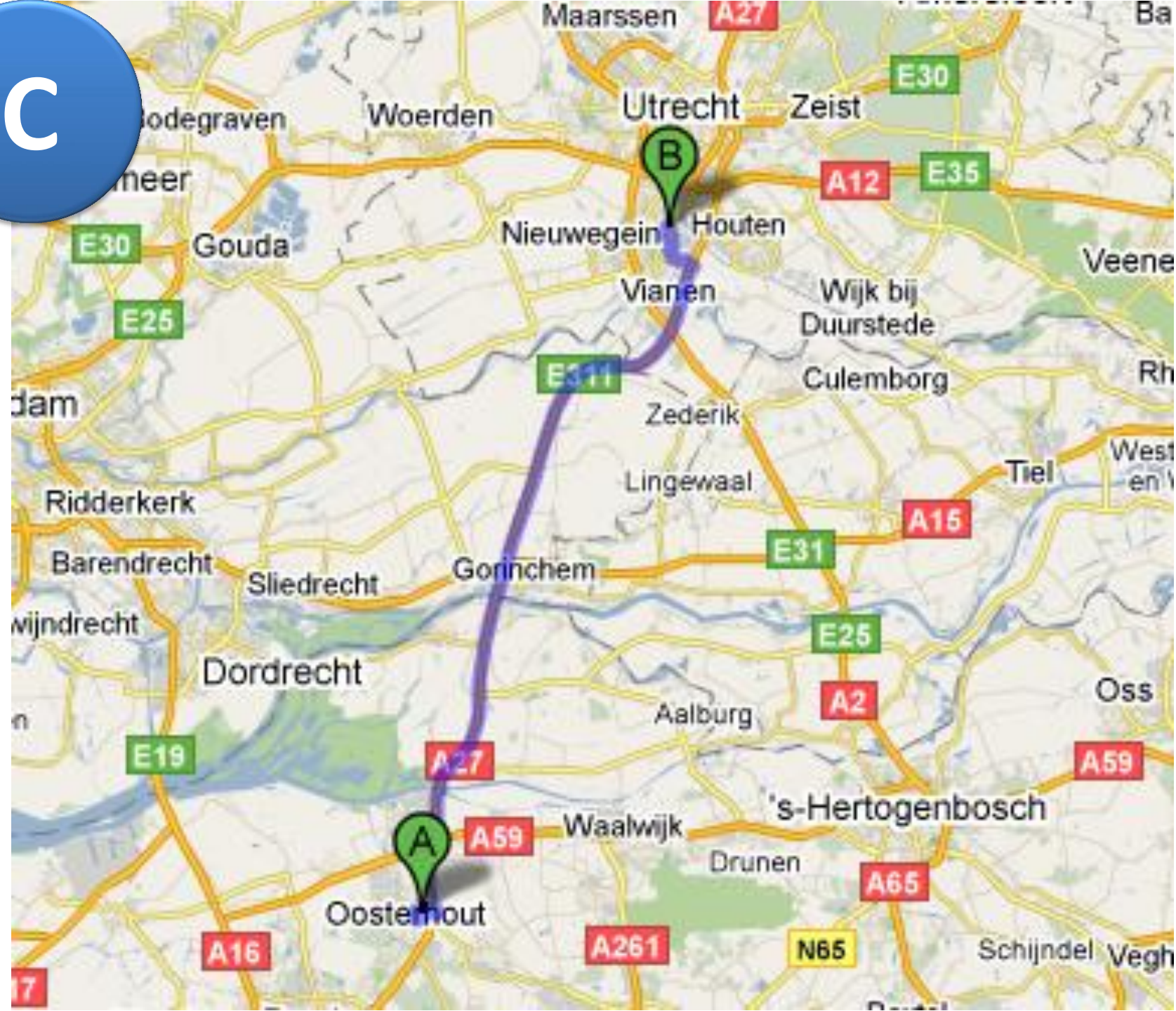
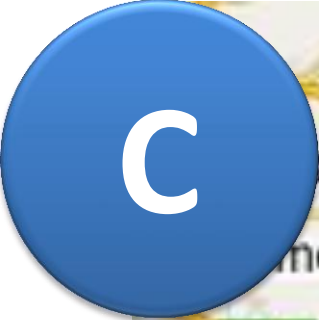


A

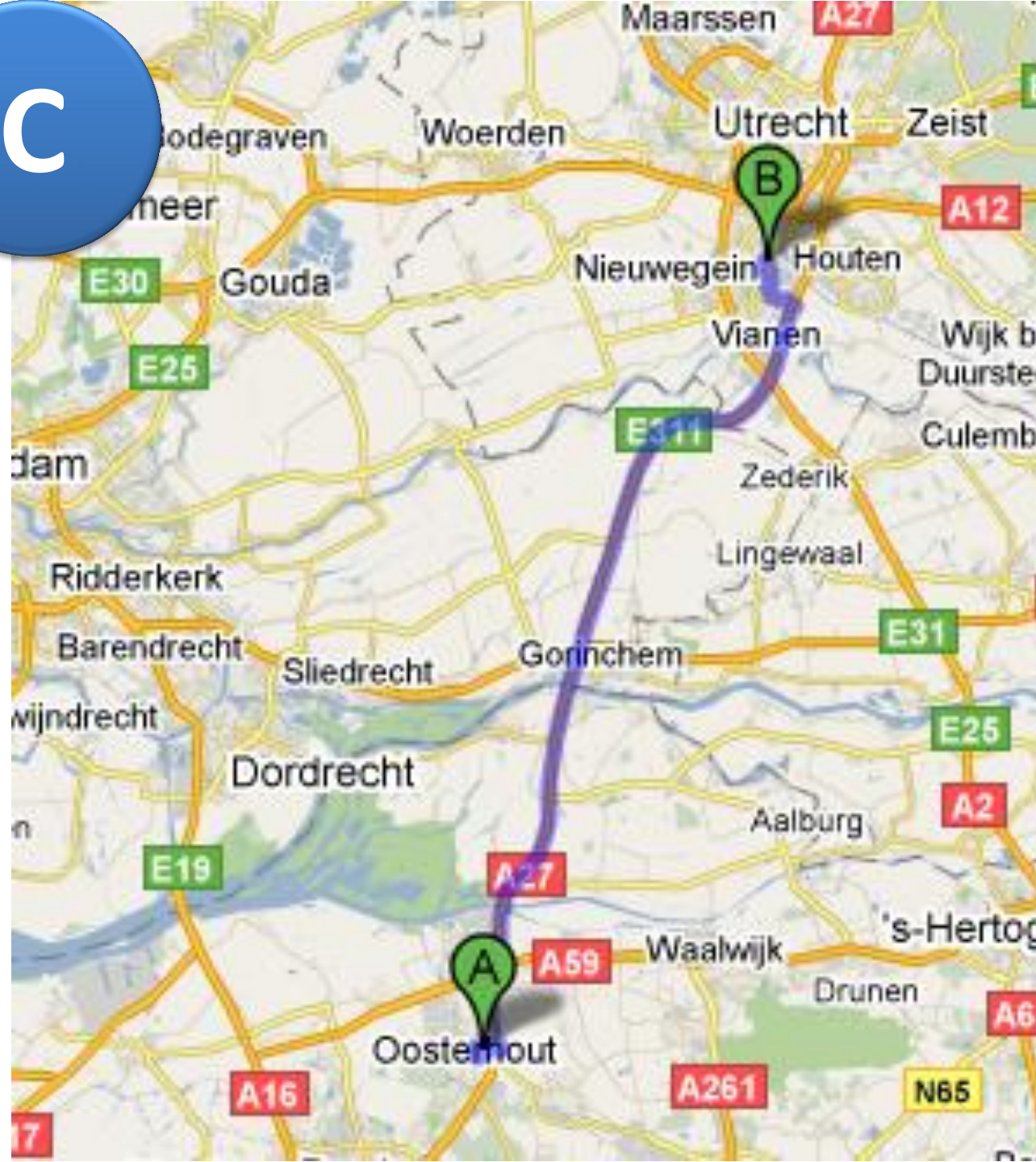
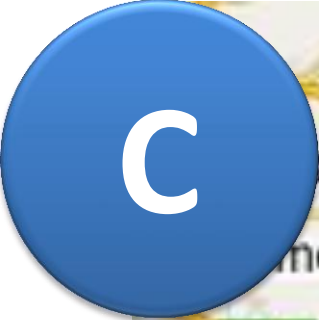


B





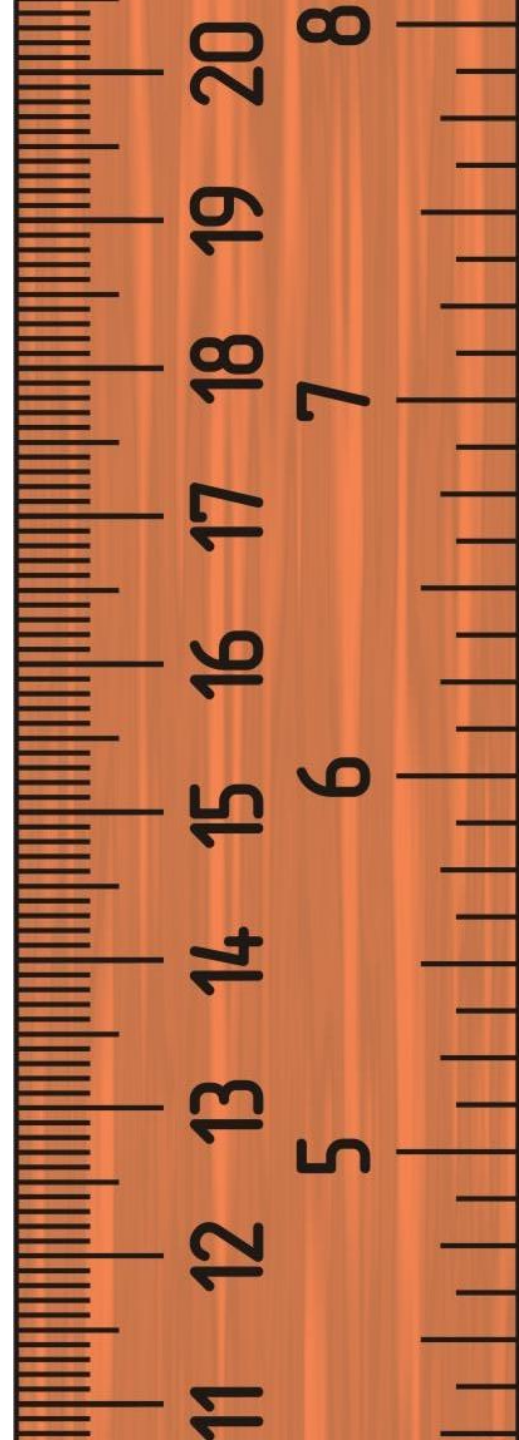




ID	DT
1	01-JAN-11
1	02-JAN-11
1	03-JAN-11
1	04-JAN-11
1	05-JAN-11
1	10-JAN-11
1	11-JAN-11
1	12-JAN-11
1	14-JAN-11
1	15-JAN-11
1	16-JAN-11



ID	DT
1	01-JAN-11
1	02-JAN-11
1	03-JAN-11
1	04-JAN-11
1	05-JAN-11
1	10-JAN-11
1	11-JAN-11
1	12-JAN-11
1	14-JAN-11
1	15-JAN-11
1	16-JAN-11



ID	DT	RN
1	01-JAN-11	1
1	02-JAN-11	2
1	03-JAN-11	3
1	04-JAN-11	4
1	05-JAN-11	5
1	10-JAN-11	6
1	11-JAN-11	7
1	12-JAN-11	8
1	14-JAN-11	9
1	15-JAN-11	10
1	16-JAN-11	11

```
SQL> select id
2         , dt
3         , row_number()
4           over (partition by id
5                 order by dt
6                 ) rn
7   from tabibitosan
8  /
```

ID	DT	RN
1	01-JAN-11	1
1	02-JAN-11	2
1	03-JAN-11	3
1	04-JAN-11	4
1	05-JAN-11	5
1	10-JAN-11	6
1	11-JAN-11	7
1	12-JAN-11	8
1	14-JAN-11	9
1	15-JAN-11	10
1	16-JAN-11	11

```
SQL> select id
2         , dt
3         - row_number()
4           over (partition by id
5                 order by dt
6                 ) grp
7   from tabibitosan
8  /
```

ID GRP

1 31-DEC-10

1 31-DEC-10

1 31-DEC-10

1 31-DEC-10

1 31-DEC-10

1 04-JAN-11

1 04-JAN-11

1 04-JAN-11

1 05-JAN-11

1 05-JAN-11

1 05-JAN-11

```
SQL> select id
2         , min (dt)
3         , max (dt)
4   from (select id
5           , dt
6           , dt
7           - row_number()
8             over (partition by id
9                   order by dt
10                  ) grp
11          from tabibitosan
12         )
13   group by id
14         , grp
15   order by id
16         , grp
17   ;
```

ID	MIN (DT)	MAX (DT)
-----	-----	-----
1	01-JAN-11	05-JAN-11
1	10-JAN-11	12-JAN-11
1	14-JAN-11	16-JAN-11
2	01-JAN-11	04-JAN-11
2	06-JAN-11	07-JAN-11
2	11-JAN-11	13-JAN-11
2	29-JAN-11	29-JAN-11



FiFo

Monty Latiolais





1700

ID	STOCKDATE	QTY
15	17-DEC-10	810
1	17-DEC-10	800
2	20-FEB-11	1600
16	20-FEB-11	1610
3	22-FEB-11	1250
17	22-FEB-11	1260
4	02-APR-11	2975

90

```
SQL> select id
      2      , stockdate
      3      , qty
      4  from fifo
      5  order by stockdate
      6  /
```

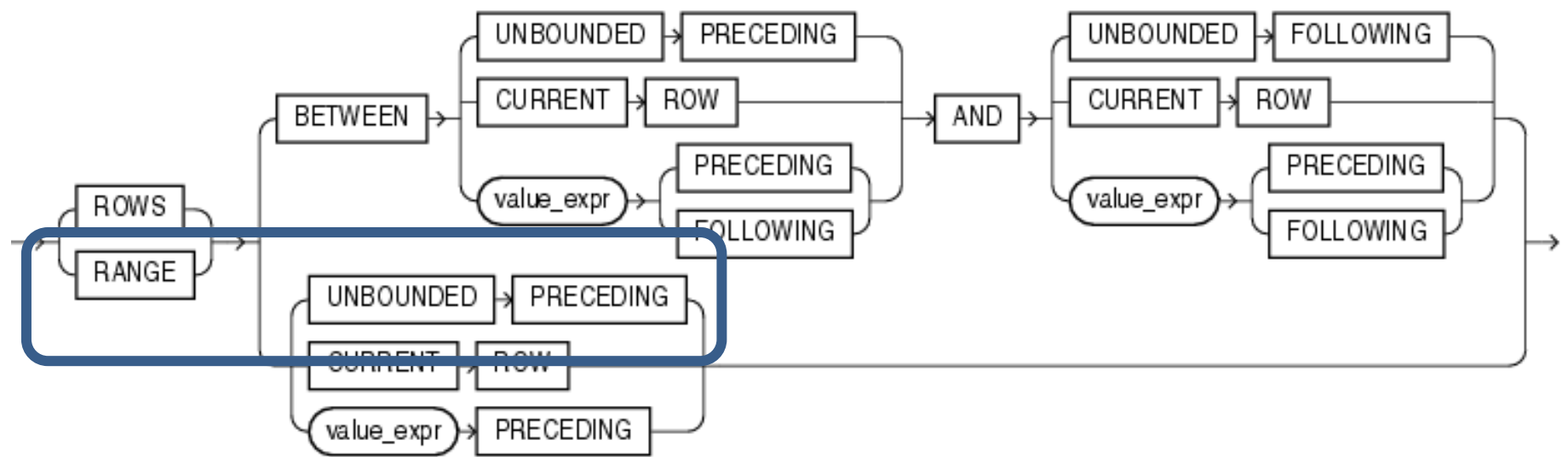
	ID	STOCKDATE	QTY
-----			-----
	15	17-DEC-10	810
	1	17-DEC-10	800
	2	20-FEB-11	1600
	16	20-FEB-11	1610
	3	22-FEB-11	1250
	17	22-FEB-11	1260
	4	02-APR-11	2975

```

SQL> select id
      2      , stockdate
      3      , qty
      4      , sum (qty)
      5          over (order by stockdate) run_ttl
      6  from fifo
      7  order by stockdate
      8  /

```

ID	STOCKDATE	QTY
15	17-DEC-10	810
1	17-DEC-10	800
2	20-FEB-11	1600
16	20-FEB-11	1610
3	22-FEB-11	1250



```

SQL> select id
      2      , stockdate
      3      , qty
      4      , sum (qty)
      5          over (order by stockdate
      6                      , id
      7                  ) run_ttl
      8  from fifo
      9  order by stockdate
     10 /

```

ID	STOCKDATE	QTY
1	17-DEC-10	800
15	17-DEC-10	810
2	20-FEB-11	1600
16	20-FEB-11	1610
3	22-FEB-11	1250


```

SQL> select id
      2      , stockdate
      3      , qty
      4      , sum (qty)
      5          over (order by stockdate
      6                  rows
      7                  between
      8                      unbounded preceding
      9                      and current row
     10                  ) run_ttl
     11 from fifo
     12 order by stockdate
     13 /

```

ID	STOCKDATE	QTY
15	17-DEC-10	810
1	17-DEC-10	800

```
SQL> select id
2         ,stockdate
3         ,qty
4         ,req_qty
5         ,req_qty -
6           sum (qty)
7             over (order by stockdate
8                   , id
9                   ) remaining_balance
10 from (select r.id
11         , f.stockdate
12         , f.qty
13         , 1700 req_qty
14         from fifo f
15         )
16 /
```

QTY	REQ_QTY	REMAINING_BALANCE
800	1700	900
810	1700	90
1600	1700	-1510

QTY	REQ_QTY	REMAINING_BALANCE
800	1700	900
810	1700	90
1600	1700	-1510

```
lag (remaining_balance
    , 1
    , req_qty
)
over (order by stockdate
      , id desc
) ship
```

QTY	REQ_QTY	REMAINING_BALANCE	SHIP
800	1700	900	1700
810	1700	90	900
1600	1700	-1510	90

```
lag (remaining_balance
    , 1
    , req_qty
)
over (order by stockdate
      , id desc
) ship
```

QTY	REQ_QTY	REMAINING_BALANCE	SHIP
800	1700	900	1700
810	1700	90	900
1600	1700	-1510	90

```
lag (remaining_balance
    , 1
    , req_qty
)
over (order by stockdate
      , id desc
) ship
```

QTY	REQ_QTY	REMAINING_BALANCE	SHIP
800	1700	900	1700
810	1700	90	900
1600	1700	-1510	90

```
least (qty, ship)
```

```
where least (qty, ship) >= 0
```

ID	STOCKDATE	QTY	LEAST (QTY , SHIP)
---	-----	-----	-----
1	17-DEC-10	800	800
15	17-DEC-10	810	810
2	20-FEB-11	1600	90

*Der er
en nemmere m
åde at gøre
det...*



Kim Berg Hansen

*There is an
easier way to
do this ...*



Kim Berg Hansen

```

SQL> select id
      2      , stockdate
      3      , qty
      4      , sum (qty) over
      5          (order by stockdate
      6              , id
      7          )
      8      ) as stock_qty
      9
     10 from fifo
     11 /

```

ID	STOCKDATE	QTY	STOCK_QTY
1	17-DEC-10	800	
15	17-DEC-10	810	800
2	20-FEB-11	1600	1610
16	20-FEB-11	1610	3210

```

SQL> select id
      2      , stockdate
      3      , qty
      4      , coalesce (
      5          sum (qty) over
      6              (order by stockdate
      7                  , id
      8                  rows between unbounded preceding
      9                      and 1 preceding
     10              )
     11      , 0) as stock_qty
     12      , 1700 req_qty
     13 from fifo
     14 /

```

	ID	STOCKDATE	QTY	STOCK_QTY	REQ_QTY
-----	-----	-----	-----	-----	-----
	1	17-DEC-10	800	0	1700
	15	17-DEC-10	810	800	1700
	2	20-FEB-11	1600	1610	1700
	16	20-FEB-11	1610	3210	1700

```

SQL> select id
      2      , stockdate
      3      , qty
      4      , stock_qty
      5      , req_qty
      6      , least (qty, req_qty - stock_qty) to_ship
      7  from
      8  (
...
22  )
23  where stock_qty < req_qty
24  ;

```

	ID	STOCKDATE	QTY	STOCK_QTY	REQ_QTY	TO_SHIP
	-----	-----	-----	-----	-----	-----
	1	17-DEC-10	800	0	1700	800
	15	17-DEC-10	810	800	1700	810
	2	20-FEB-11	1600	1610	1700	90

Requirement:
“Find the most recent contract.”



```

select id
      , cat
      , start_date
      , end_date
  from contracts main_q
 where not exists
(select 1
  from contracts nest_q
 where nest_q.cat = main_q.cat
    and Decode (nest_q.end_date, main_q.end_date
                , Decode (nest_q.start_date, main_q.start_date
                        , Trunc (sysdate, 'DDD')
                        +
                        Decode (nest_q.id
                              , Least (nest_q.id, main_q.id)
                              , 0, 1)
                        , Nvl(nest_q.start_date
                            , main_q.start_date-1)
                        )
                , Nvl(nest_q.end_date, main_q.end_date+1))
    >
    Decode (main_q.end_date, nest_q.end_date
          , Decode (main_q.start_date, nest_q.start_date
                  , Trunc (sysdate, 'DDD')
                  +
                  Decode (main_q.id
                        , Least (nest_q.id, main_q.id)
                        , 0, 1)
                  , Nvl(main_q.start_date
                      , nest_q.start_date-1)
                  )
          , Nvl(main_q.end_date, nest_q.end_date+1))
    )
)

```

Requirement:

“Find the most recent contract.”

- Highest End date, nulls are in the future.
- Highest Start date, nulls are to be ignored.
- Highest ID (Arbitrarily)
- *Per Category*




```

select id
      , cat
      , start_date
      , end_date
  from contracts main_q
 where not exists
 (select 1
   from contracts nest_q
  where nest_q.cat = main_q.cat
        and Decode (nest_q.end_date, main_q.end_date
                    , Decode (nest_q.start_date, main_q.start_date
                              , Trunc (sysdate, 'DDD')
                              +
                              Decode (nest_q.id
                                     , Least (nest_q.id, main_q.id)
                                     , 0, 1)
                              , Nvl(nest_q.start_date
                                   , main_q.start_date-1)
                              )
                    , Nvl(nest_q.end_date, main_q.end_date+1))
        >
        Decode (main_q.end_date, nest_q.end_date
                , Decode (main_q.start_date, nest_q.start_date
                          , Trunc (sysdate, 'DDD')
                          +
                          Decode (main_q.id
                                 , Least (nest_q.id, main_q.id)
                                 , 0, 1)
                          , Nvl(main_q.start_date
                               , nest_q.start_date-1)
                          )
                , Nvl(main_q.end_date, nest_q.end_date+1))
        )
  )

```

Compare End Date

Compare Start Date

Compare ID

The Analytic Way

- First Inclination
 - *Per Category* → *Partition by Category*
- Which Analytic Function?
 - Lead() ?
 - Lag() ?
 - Max() ?
 - ...?







Rephrase Requirement





**“Most Recent
Contract”**



**“Top One
Contract”**



Ranking

Top One Contracts

```
select id
      , cat
      , start_date
      , end_date
from (select id
          , cat
          , start_date
          , end_date
          , Row_Number() over (
                                partition by cat
                                order by end_date desc nulls first
                                      , start_date desc nulls last
                                      , id desc
                              ) rn
      from contracts
    )
where rn = 1
```

From the Requirement:

- Per Category
- Highest End date, nulls are in the future.
- Highest Start date, nulls are to be ignored.
- Highest ID (Arbitrarily)

Execution Time

- Original Query: 00:00:00.01
 - Supposed to be *slower*
- Analytic Query: 00:00:00.03
 - Supposed to be *faster*
- What about the statistics?

Statistics 60 Records

Statistics

```
-----  
259 recursive calls  
0 db block gets  
165 consistent gets  
0 physical reads  
0 redo size  
1217 bytes sent via SQL*Net to client  
519 bytes received via SQL*Net from client  
3 SQL*Net roundtrips to/from client  
6 sorts (memory)  
0 sorts (disk)  
30 rows processed
```

Original Query

Analytic Query

Statistics

```
-----  
1 recursive calls  
0 db block gets  
7 consistent gets  
0 physical reads  
0 redo size  
1217 bytes sent via SQL*Net to client  
519 bytes received via SQL*Net from client  
3 SQL*Net roundtrips to/from client  
1 sorts (memory)  
0 sorts (disk)  
30 rows processed
```

Statistics 7.680 Records

Statistics

```
-----  
42 recursive calls  
0 db block gets  
71646 consistent gets  
0 physical reads  
0 redo size  
1247 bytes sent via SQL*Net to client  
519 bytes received via SQL*Net from client  
3 SQL*Net roundtrips to/from client  
0 sorts (memory)  
0 sorts (disk)  
30 rows processed
```

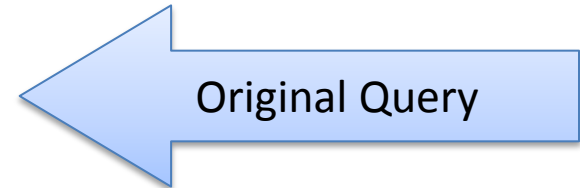
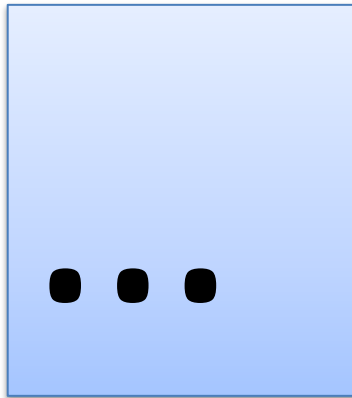
Original Query

Analytic Query

Statistics

```
-----  
0 recursive calls  
0 db block gets  
31 consistent gets  
0 physical reads  
0 redo size  
1247 bytes sent via SQL*Net to client  
519 bytes received via SQL*Net from client  
3 SQL*Net roundtrips to/from client  
1 sorts (memory)  
0 sorts (disk)  
30 rows processed
```

Statistics 1.966.080 Records



Statistics

```
-----  
1 recursive calls  
88 db block gets  
6087 consistent gets  
6120 physical reads  
664 redo size  
1247 bytes sent via SQL*Net to client  
519 bytes received via SQL*Net from client  
3 SQL*Net roundtrips to/from client  
0 sorts (memory)  
1 sorts (disk)  
30 rows processed
```

Timing Results

Record Count	Original Query	Analytic Query
60	00:00:00.01	00:00:00.03
7.680	00:00:01.65	00:00:00.07
1.966.080		00:00:08.37

... but is an analytic function **really** necessary?

Joins,
WHERE, GROUP BY,
and HAVING clauses



Partitions created;
Analytic functions
applied to each row in
each partition

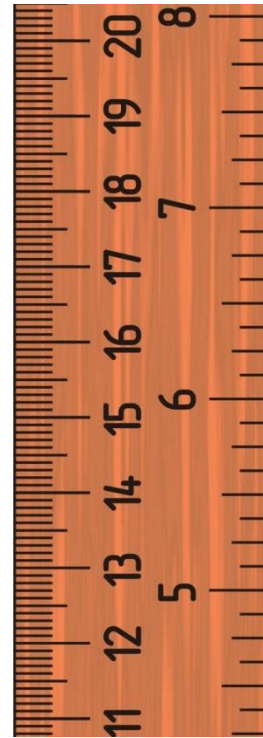


Final
ORDER BY

```
select max (id)
      , cat
      , max (start_date)
          keep (dense_rank first
                order by end_date desc nulls first
                      , start_date desc nulls last
                      , id desc
                ) start_date
      , max (end_date)
          keep (dense_rank first
                order by end_date desc nulls first
                      , start_date desc nulls last
                      , id desc
                ) end_date
from contracts
group by cat
```

Not in this case...

Filling in the
GAP



**“Best thing since
SELECT statement”**

Tom Kyte



COMMITTED TO ICT.
INVOLVED IN PEOPLE.

Alex Nuijten
alex.nuijten@amis.nl

Technology.amis.nl
Nuijten.blogspot.com
@alexnuijten



omroepstaphorst.nl

marchismind.wordpress.com/2011/04/10/the-gap-is-hell/



www.gap.com

旅人さん

www.google.com



www.wtmrac.com/Sticky-Note.png



jamesjames.web-log.nl



montylatiolais.blogspot.com



www.nokia.com

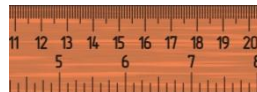
nationalgeographic.com



www.adpic.de



www.middlesexbank.com



mrszechman.acsd.wikispaces.net/Math+Cover+Page



www.amsterdamimage.com