

# Analytic Functions: Unleash the SQL Power



Alex Nuijten



# Analytic Functions: Unleash the SQL Power



Alex Nuijten





**WARNING**



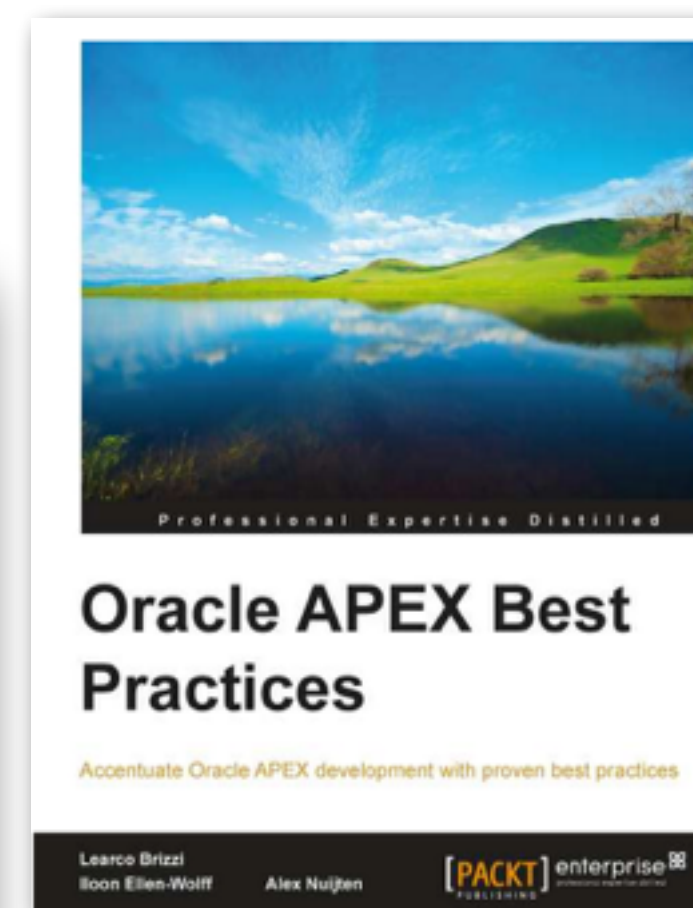
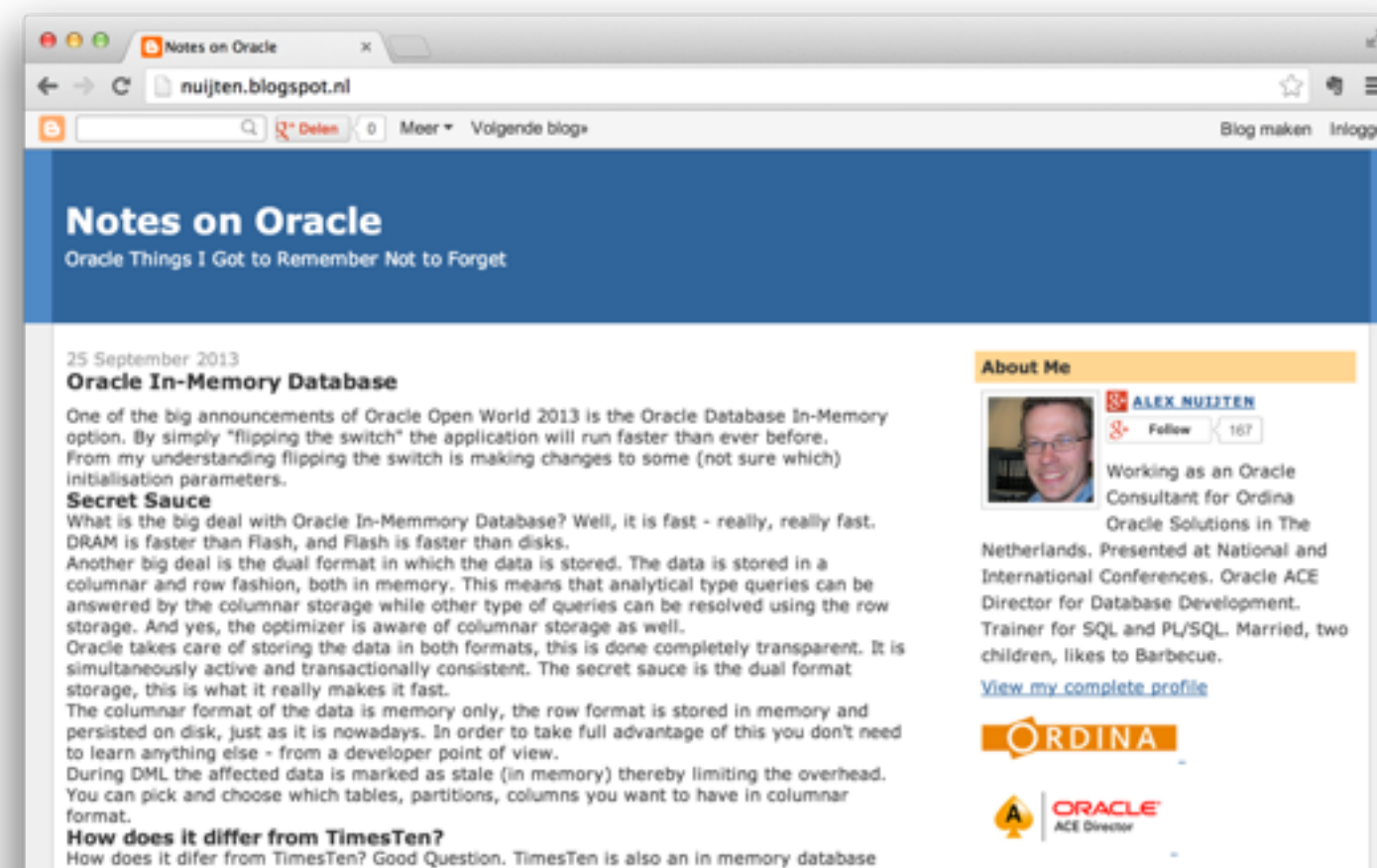
**lots and lots**

There will be code in this  
presentation

```
select 'can you read this?'  
from dual;
```



nuijten.blogspot.com



**ORACLE<sup>®</sup>**  

---

**DATABASE** **12<sup>c</sup>**

**ORACLE<sup>®</sup>**  

---

**DATABASE** **11<sup>g</sup>**

**ORACLE<sup>®</sup>**  

---

**DATABASE** **10<sup>g</sup>**









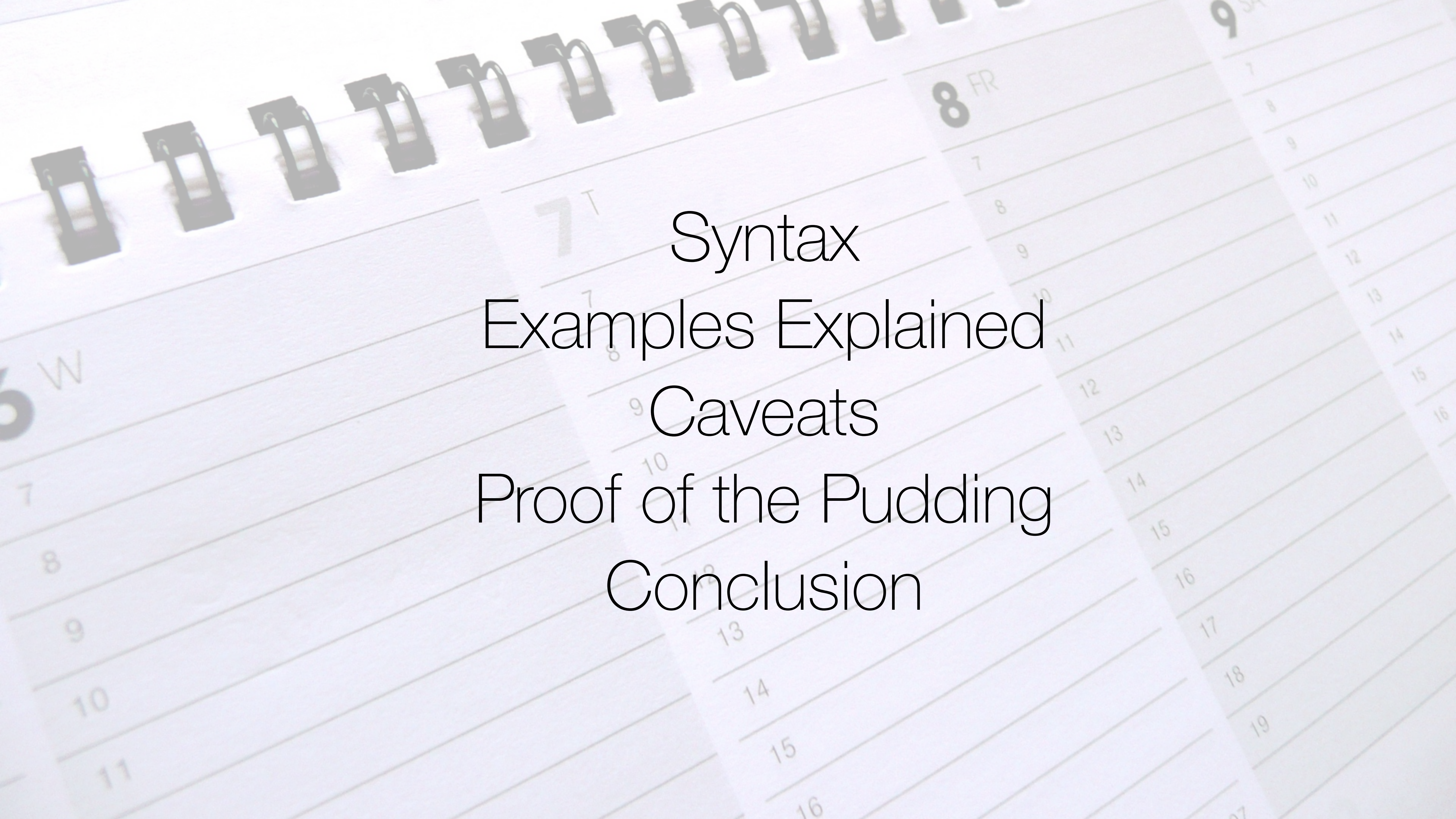
8.1.6 EE



1999

"Best thing since SELECT statement"

*Tom Kyte*

A spiral-bound notebook with a calendar grid is shown in the background. The notebook is open to a page with a grid of dates. The dates are arranged in columns, with the first column starting with '7 T', the second with '8 FR', and the third with '9 SA'. The numbers 7 through 19 are visible in the grid. The notebook has a silver spiral binding on the left side.

Syntax  
Examples Explained  
Caveats  
Proof of the Pudding  
Conclusion



View 125% Formula Table Chart Text Shape Media Comment Share Tips Format Sort & Filter

+ Sheet 1

A B C

| Employees |        |        |      |
|-----------|--------|--------|------|
| 1         | ENAME  | DEPTNO | SAL  |
| 2         | CLARK  | 10     | 2450 |
| 3         | KING   | 10     | 5000 |
| 4         | MILLER | 10     | 1300 |
| 5         | ADAMS  | 20     | 1100 |
| 6         | FORD   | 20     | 3000 |
| 7         | JONES  | 20     | 2975 |
| 8         | SCOTT  | 20     | 3000 |
| 9         | SMITH  | 20     | 800  |
| 10        | ALLEN  | 30     | 1600 |
| 11        | BLAKE  | 30     | 2850 |
| 12        | JAMES  | 30     | 950  |
| 13        | MARTIN | 30     | 1250 |
| 14        | TURNER | 30     | 1500 |
| 15        | WARD   | 30     | 1250 |

=

View 125% Formula Table Chart Text Shape Media Comment Share Tips Format Sort & Filter

+ Sheet 1

A B C D E F

| Employees |        |        |      |               |                  |     |
|-----------|--------|--------|------|---------------|------------------|-----|
| 1         | ENAME  | DEPTNO | SAL  | Running Total | Department Total | Seq |
| 2         | CLARK  | 10     | 2450 | 2450          | 2450             | 1   |
| 3         | KING   | 10     | 5000 | 7450          | 7450             | 2   |
| 4         | MILLER | 10     | 1300 | 8750          | 8750             | 3   |
| 5         | ADAMS  | 20     | 1100 | 9850          | 1100             | 1   |
| 6         | FORD   | 20     | 3000 | 12850         | 4100             | 2   |
| 7         | JONES  | 20     | 2975 | 15825         | 7075             | 3   |
| 8         | SCOTT  | 20     | 3000 | 18825         | 10075            | 4   |
| 9         | SMITH  | 20     | 800  | 19625         | 10875            | 5   |
| 10        | ALLEN  | 30     | 1600 | 21225         | 1600             | 1   |
| 11        | BLAKE  | 30     | 2850 | 24075         | 4450             | 2   |
| 12        | JAMES  | 30     | 950  | 25025         | 5400             | 3   |
| 13        | MARTIN | 30     | 1250 | 26275         | 6650             | 4   |
| 14        | TURNER | 30     | 1500 | 27775         | 8150             | 5   |
| 15        | WARD   | 30     | 1250 | 29025         | 9400             | 6   |

=





SQL> @analytic

| ENAME  | DEPTNO | SAL  | RUNNING_TOTAL | DEPARTMENT_TOTAL | SEQ |
|--------|--------|------|---------------|------------------|-----|
| CLARK  | 10     | 2450 | 2450          | 2450             | 1   |
| KING   | 10     | 5000 | 7450          | 7450             | 2   |
| MILLER | 10     | 1300 | 8750          | 8750             | 3   |
| ADAMS  | 20     | 1100 | 9850          | 1100             | 1   |
| FORD   | 20     | 3000 | 12850         | 4100             | 2   |
| JONES  | 20     | 2975 | 15825         | 7075             | 3   |
| SCOTT  | 20     | 3000 | 18825         | 10075            | 4   |
| SMITH  | 20     | 800  | 19625         | 10875            | 5   |
| ALLEN  | 30     | 1600 | 21225         | 1600             | 1   |
| BLAKE  | 30     | 2850 | 24075         | 4450             | 2   |
| JAMES  | 30     | 950  | 25025         | 5400             | 3   |
| MARTIN | 30     | 1250 | 26275         | 6650             | 4   |
| TURNER | 30     | 1500 | 27775         | 8150             | 5   |
| WARD   | 30     | 1250 | 29025         | 9400             | 6   |

14 rows selected.

SQL> █

```
SQL> select e.ename
2         ,e.deptno
3         ,e.sal
4         ,(select sum (e2.sal)
5             from emp e2
6             where e2.deptno < e.deptno
7                 or (e2.deptno = e.deptno
8                     and e2.ename <= e.ename
9                 )
10        ) running_total
11        ,(select sum (e3.sal)
12            from emp e3
13            where e3.deptno = e.deptno
14                and e3.ename <= e.ename
15        ) department_total
16        ,(select count(e4.ename)
17            from emp e4
18            where e4.deptno = e.deptno
19                and e4.ename <= e.ename
20        ) seq
21    from emp e
22    order by e.deptno
23            ,e.ename
24    /
```

```
SQL> select e.ename
2         ,e.deptno
3         ,e.sal
4         ,(select sum (e2.sal)
5             from emp e2
6             where e2.deptno < e.deptno
7                 or (e2.deptno = e.deptno
8                     and e2.ename <= e.ename
9                 )
10        ) running_total
11        ,(select sum (e3.sal)
12            from emp e3
13            where e3.deptno = e.deptno
14                and e3.ename <= e.ename
15        ) department_total
16        ,(select count(e4.ename)
17            from emp e4
18            where e4.deptno = e.deptno
19                and e4.ename <= e.ename
20        ) seq
21    from emp e
22    order by e.deptno
23            ,e.ename
24    /
```

```
SQL> select e.ename
2         ,e.deptno
3         ,e.sal
4         ,(select sum (e2.sal)
5             from emp e2
6             where e2.deptno < e.deptno
7                 or (e2.deptno = e.deptno
8                     and e2.ename <= e.ename
9                 )
10        ) running_total
11        ,(select sum (e3.sal)
12            from emp e3
13            where e3.deptno = e.deptno
14                and e3.ename <= e.ename
15        ) department_total
16        ,(select count(e4.ename)
17            from emp e4
18            where e4.deptno = e.deptno
19                and e4.ename <= e.ename
20        ) seq
21    from emp e
22    order by e.deptno
23            ,e.ename
24    /
```

```
SQL> select e.ename
2      ,e.deptno
3      ,e.sal
4      ,(select sum (e2.sal)
5          from emp e2
6          where e2.deptno < e.deptno
7              or (e2.deptno = e.deptno
8                  and e2.ename <= e.ename
9              )
10     ) running_total
11     ,(select sum (e3.sal)
12         from emp e3
13         where e3.deptno = e.deptno
14             and e3.ename <= e.ename
15     ) department_total
16     ,(select count(e4.ename)
17         from emp e4
18         where e4.deptno = e.deptno
19         and e4.ename <= e.ename
20     ) seq
21     from emp e
22     order by e.deptno
23             ,e.ename
24 /
```

```
SQL> select e.ename
2      ,e.deptno
3      ,e.sal
4      ,(select sum (e2.sal)
5          from emp e2
6          where e2.deptno < e.deptno
7              or (e2.deptno = e.deptno
8                  and e2.ename <= e.ename
9              )
10     ) running_total
11     ,(select sum (e3.sal)
12         from emp e3
13         where e3.deptno = e.deptno
14             and e3.ename <= e.ename
15     ) department_total
16     ,(select count (e4.ename)
17         from emp e4
18         where e4.deptno = e.deptno
19             and e4.ename <= e.ename
20     ) seq
21     from emp e
22     order by e.deptno
23            ,e.ename
24 /
```

```
SQL> select e.ename
2         ,e.deptno
3         ,e.sal
4         ,sum (e.sal) over (order by e.deptno, e.ename) running_total
5         ,sum (e.sal) over (partition by e.deptno
6                             order by e.deptno, e.ename) department_total
7         ,row_number() over (partition by e.deptno
8                             order by e.ename) seq
9     from emp e
10 /
```

```
SQL> select e.ename
2         ,e.deptno
3         ,e.sal
4         ,sum (e.sal) over (order by e.deptno, e.ename) running_total
5         ,sum (e.sal) over (partition by e.deptno
6                             order by e.deptno, e.ename) department_total
7         ,row_number() over (partition by e.deptno
8                             order by e.ename) seq
9     from emp e
10 /
```



```
SQL> select e.ename
2         ,e.deptno
3         ,e.sal
4         ,sum (e.sal) over (order by e.deptno, e.ename) running_total
5         ,sum (e.sal) over (partition by e.deptno
6         order by e.deptno, e.ename) department_total
7         ,row_number() over (partition by e.deptno
8         order by e.ename) seq
9     from emp e
10 /
```

```
SQL> select e.ename
2         ,e.deptno
3         ,e.sal
4         ,sum (e.sal) over (order by e.deptno, e.ename) running_total
5         ,sum (e.sal) over (partition by e.deptno
6                             order by e.deptno, e.ename) department_total
7         ,row_number() over (partition by e.deptno
8                             order by e.ename) seq
9     from emp e
10 /
```

```
SQL> select e.ename
2         ,e.deptno
3         ,e.sal
4         ,sum (e.sal) over (order by e.deptno, e.ename) running_total
5         ,sum (e.sal) over (partition by e.deptno
6                             order by e.deptno, e.ename) department_total
7         ,row_number() over (partition by e.deptno
8                             order by e.ename) seq
9 from emp e
10 /
```

# Functions

COUNT

MIN

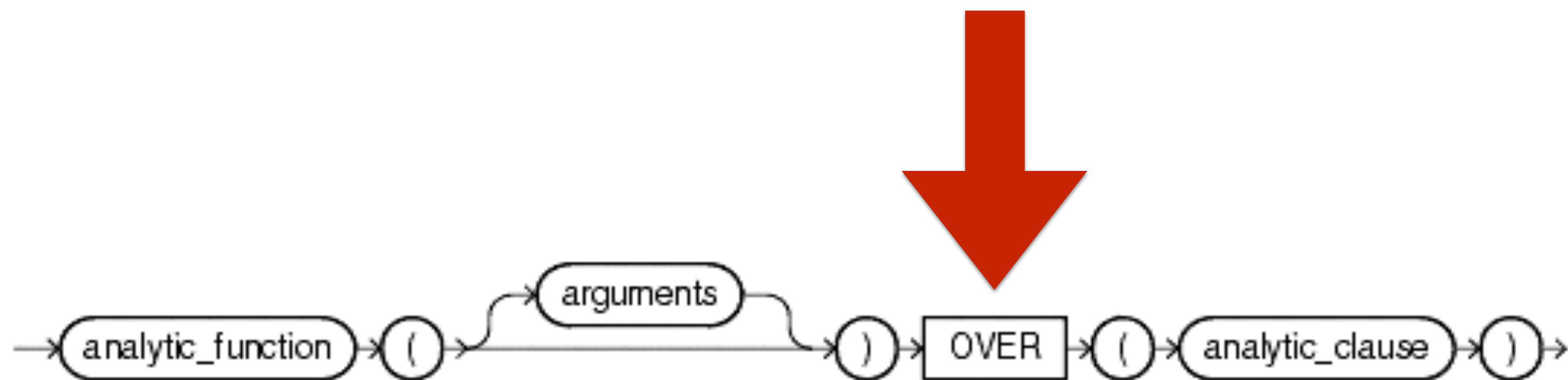
MAX

SUM

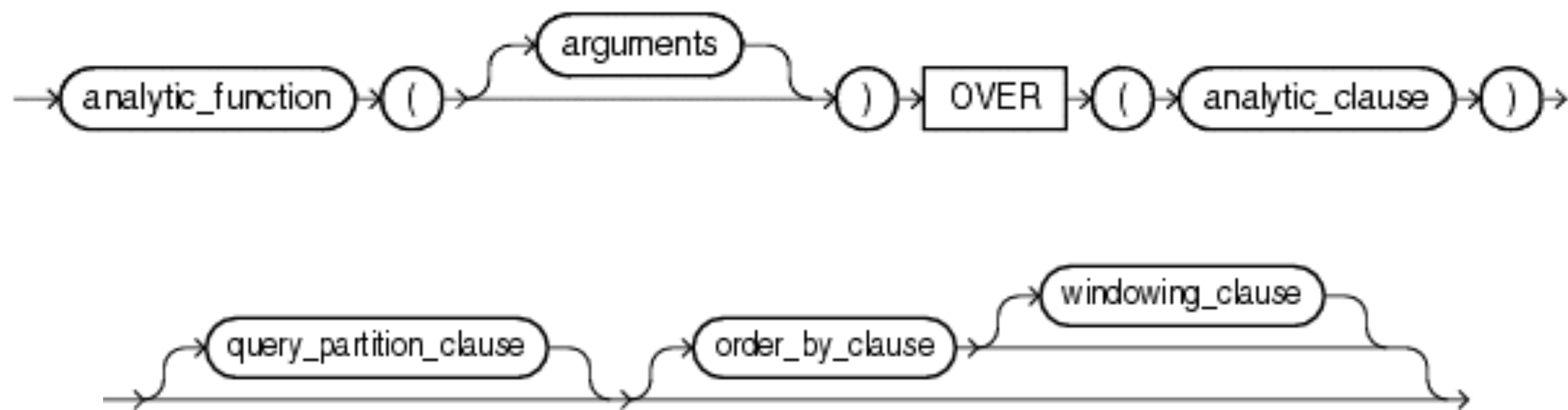
AVG

NTH\_VALUE  
LISTAGG      NTILE      LAG  
RANK      COUNT      LEAD  
          MIN      MAX  
DENSE\_RANK      SUM      AVG      LAST\_VALUE  
ROW\_NUMBER      FIRST\_VALUE  
RATIO\_TO\_REPORT

PERCENTILE\_DISC  
PERCENTILE\_CONT  
PERCENTILE\_RANK  
VAR\_SAMP  
STDDEV\_SAMP  
MEDIAN  
CLUSTER\_DETAILS  
CLUSTER\_DISTANCE  
CLUSTER\_ID  
REGR\_SLOPE  
REGR\_INTERCEPT  
PREDICTION\_COST  
PREDICTION  
VAR\_POP  
LISTAGG  
RANK  
DENSE\_RANK  
CORR  
CLUSTER\_PROBABILITY  
PREDICTION\_COST  
PREDICTION  
NTH\_VALUE  
COUNT  
MIN  
SUM  
RATIO\_TO\_REPORT  
FEATURE\_ID  
REGR\_COUNT  
PREDICTION\_PROBABILITY  
PREDICTION\_DETAILS  
LAG  
LEAD  
LAST\_VALUE  
REGR\_AVGX  
REGR\_SXX  
REGR\_R2  
PREDICTION\_SET  
CLUSTER\_SET  
VARIANCE  
COVAR\_SAMP  
FIRST  
LAST  
FEATURE\_VALUE  
FEATURE\_DETAILS  
FEATURE\_SET  
REGR\_SYY  
REGR\_AVGY  
REGR\_SXY



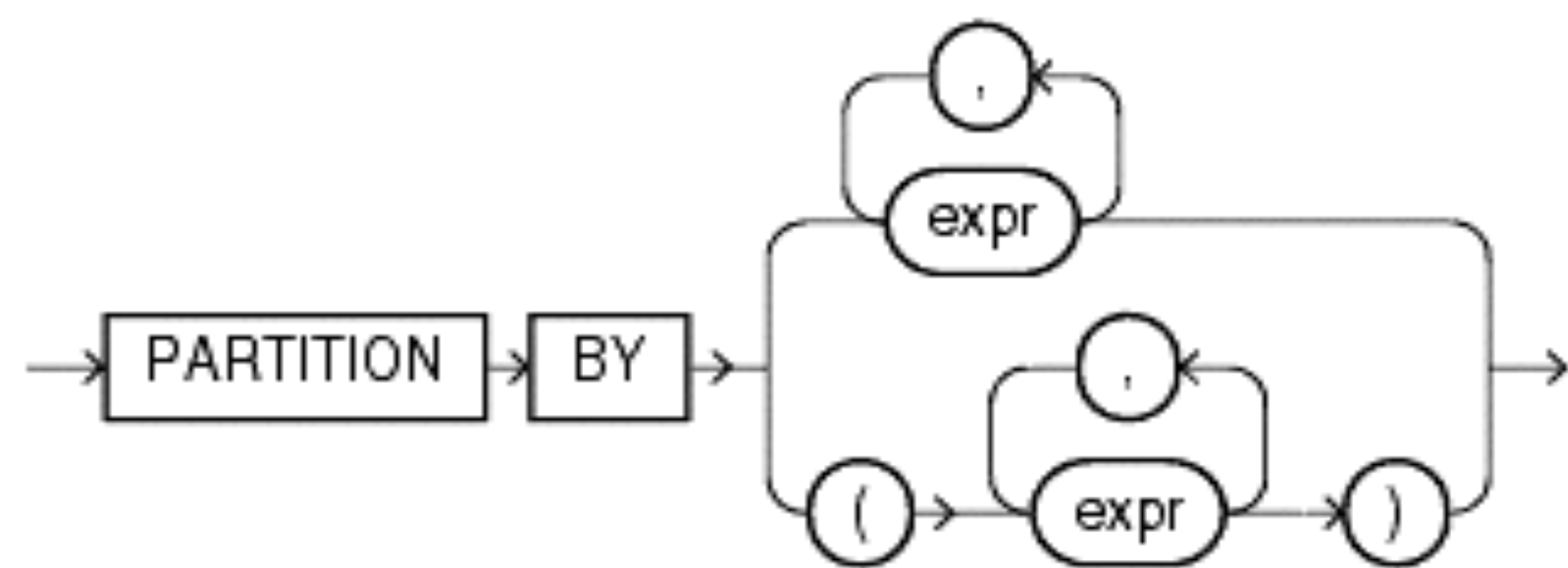




Example

```
SQL> select ename
2      , deptno
3      , job
4      , sal
5      , avg (sal) over () avg_sal
6 from emp
7 order by deptno
8 ;
```

| ENAME  | DEPTNO | JOB       | SAL  | AVG_SAL    |
|--------|--------|-----------|------|------------|
| CLARK  | 10     | MANAGER   | 2450 | 2073.21429 |
| KING   | 10     | PRESIDENT | 5000 | 2073.21429 |
| MILLER | 10     | CLERK     | 1300 | 2073.21429 |
| JONES  | 20     | MANAGER   | 2975 | 2073.21429 |
| FORD   | 20     | ANALYST   | 3000 | 2073.21429 |
| ADAMS  | 20     | CLERK     | 1100 | 2073.21429 |
| SMITH  | 20     | CLERK     | 800  | 2073.21429 |
| SCOTT  | 20     | ANALYST   | 3000 | 2073.21429 |
| WARD   | 30     | SALESMAN  | 1250 | 2073.21429 |
| TURNER | 30     | SALESMAN  | 1500 | 2073.21429 |
| ALLEN  | 30     | SALESMAN  | 1600 | 2073.21429 |
| JAMES  | 30     | CLERK     | 950  | 2073.21429 |
| BLAKE  | 30     | MANAGER   | 2850 | 2073.21429 |
| MARTIN | 30     | SALESMAN  | 1250 | 2073.21429 |



```
SQL> select ename
2      , deptno
3      , job
4      , sal
5      , avg (sal) over () avg_sal
6 from emp
7 order by deptno
8 ;
```

| ENAME  | DEPTNO | JOB      | SAL  | AVG_SAL    |
|--------|--------|----------|------|------------|
| CLARK  |        |          | 450  | 2073.21429 |
| KING   |        |          | 5000 | 2073.21429 |
| MILLER |        |          | 300  | 2073.21429 |
| JONES  |        |          | 2975 | 2073.21429 |
| FORD   | 20     | ANALYST  | 3000 | 2073.21429 |
| ADAMS  | 20     | CLERK    | 1100 | 2073.21429 |
| SMITH  | 20     | CLERK    | 800  | 2073.21429 |
| SCOTT  | 20     | ANALYST  | 3000 | 2073.21429 |
| WARD   | 30     | SALESMAN | 1250 | 2073.21429 |
| TURNER | 30     | SALESMAN | 1500 | 2073.21429 |
| ALLEN  | 30     | SALESMAN | 1600 | 2073.21429 |
| JAMES  | 30     | CLERK    | 950  | 2073.21429 |
| BLAKE  | 30     | MANAGER  | 2850 | 2073.21429 |
| MARTIN | 30     | SALESMAN | 1250 | 2073.21429 |



No Partition Clause

```
SQL> select ename
2      , deptno
3      , job
4      , sal
5      , avg (sal) over () avg_sal
6 from emp
7 order by deptno
8 ;
```

| ENAME  | DEPTNO | JOB       | SAL  | AVG_SAL    |
|--------|--------|-----------|------|------------|
| CLARK  | 10     | MANAGER   | 2450 | 2073.21429 |
| KING   | 10     | PRESIDENT | 5000 | 2073.21429 |
| MILLER | 10     | CLERK     | 1300 | 2073.21429 |
| JONES  | 20     | MANAGER   | 2975 | 2073.21429 |
| FORD   | 20     | ANALYST   | 3000 | 2073.21429 |
| ADAMS  | 20     | CLERK     | 1100 | 2073.21429 |
| SMITH  | 20     | CLERK     | 800  | 2073.21429 |
| SCOTT  | 20     | ANALYST   | 3000 | 2073.21429 |
| WARD   | 30     | SALESMAN  | 1250 | 2073.21429 |
| TURNER | 30     | SALESMAN  | 1500 | 2073.21429 |
| ALLEN  | 30     | SALESMAN  | 1600 | 2073.21429 |
| JAMES  | 30     | CLERK     | 950  | 2073.21429 |
| BLAKE  | 30     | MANAGER   | 2850 | 2073.21429 |
| MARTIN | 30     | SALESMAN  | 1250 | 2073.21429 |

```
SQL> select ename
2      , deptno
3      , job
4      , sal
5      , avg (sal) over (partition by deptno) avg_dept_sal
6 from emp
7 order by deptno
8 ;
```

| ENAME  | DEPTNO | JOB       | SAL  | AVG_DEPT_SAL |
|--------|--------|-----------|------|--------------|
| CLARK  | 10     | MANAGER   | 2450 | 2916.66667   |
| KING   | 10     | PRESIDENT | 5000 | 2916.66667   |
| MILLER | 10     | CLERK     | 1300 | 2916.66667   |
| JONES  | 20     | MANAGER   | 2975 | 2175         |
| FORD   | 20     | ANALYST   | 3000 | 2175         |
| ADAMS  | 20     | CLERK     | 1100 | 2175         |
| SMITH  | 20     | CLERK     | 800  | 2175         |
| SCOTT  | 20     | ANALYST   | 3000 | 2175         |
| WARD   | 30     | SALESMAN  | 1250 | 1566.66667   |
| TURNER | 30     | SALESMAN  | 1500 | 1566.66667   |
| ALLEN  | 30     | SALESMAN  | 1600 | 1566.66667   |
| JAMES  | 30     | CLERK     | 950  | 1566.66667   |
| BLAKE  | 30     | MANAGER   | 2850 | 1566.66667   |
| MARTIN | 30     | SALESMAN  | 1250 | 1566.66667   |

# Resultset



# Resultset

Partition

Partition

# Resultset

Partition

Window

Partition

Resultset

Partition

Window

Resultset

Partition

Window

Resultset

Partition

Window

Resultset

Partition

Window

Resultset

Partition

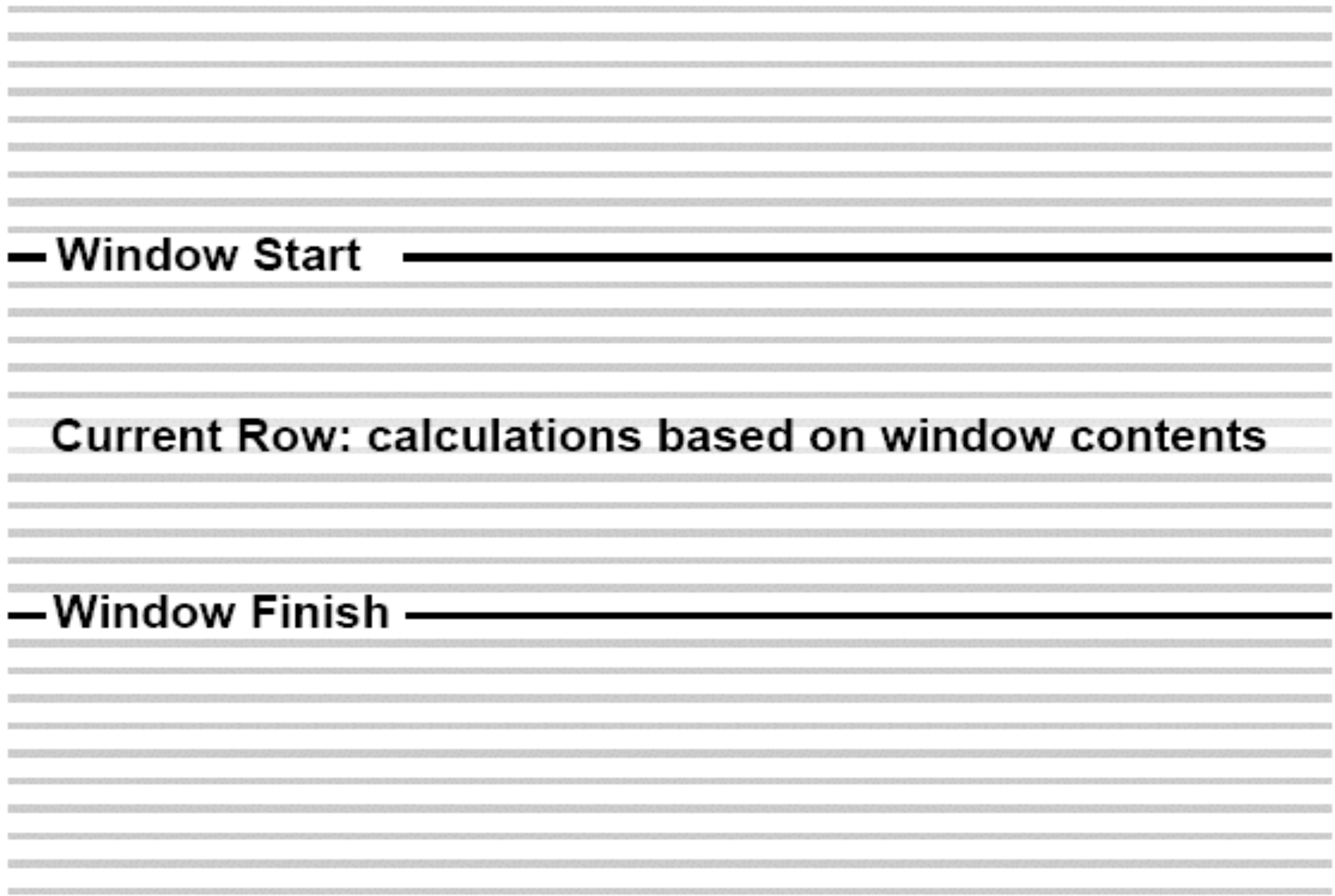
Window

Resultset

Partition

Window



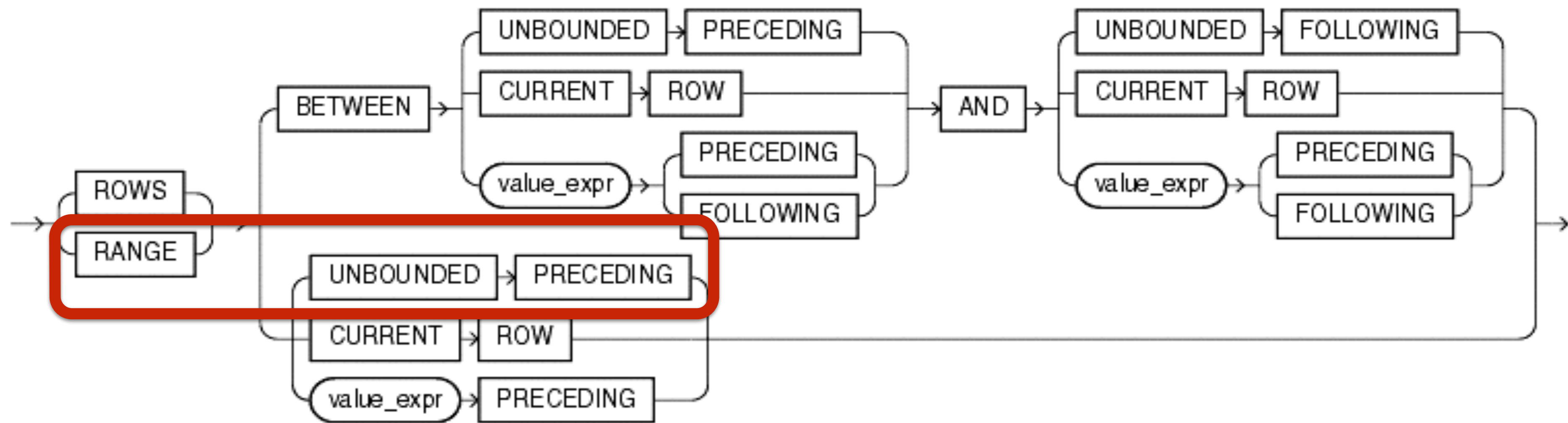
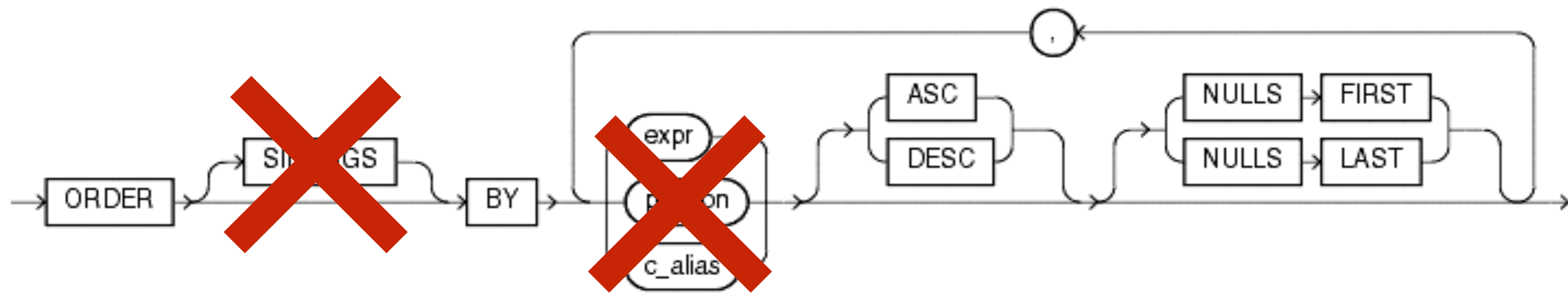


## ROWS: Number of Rows

rows between 2 preceding and 4 following

## RANGE: Numeric Offset

range between 200 preceding and 175 following



$$1 + 2 + 3$$

$$3 + 2 + 1$$

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

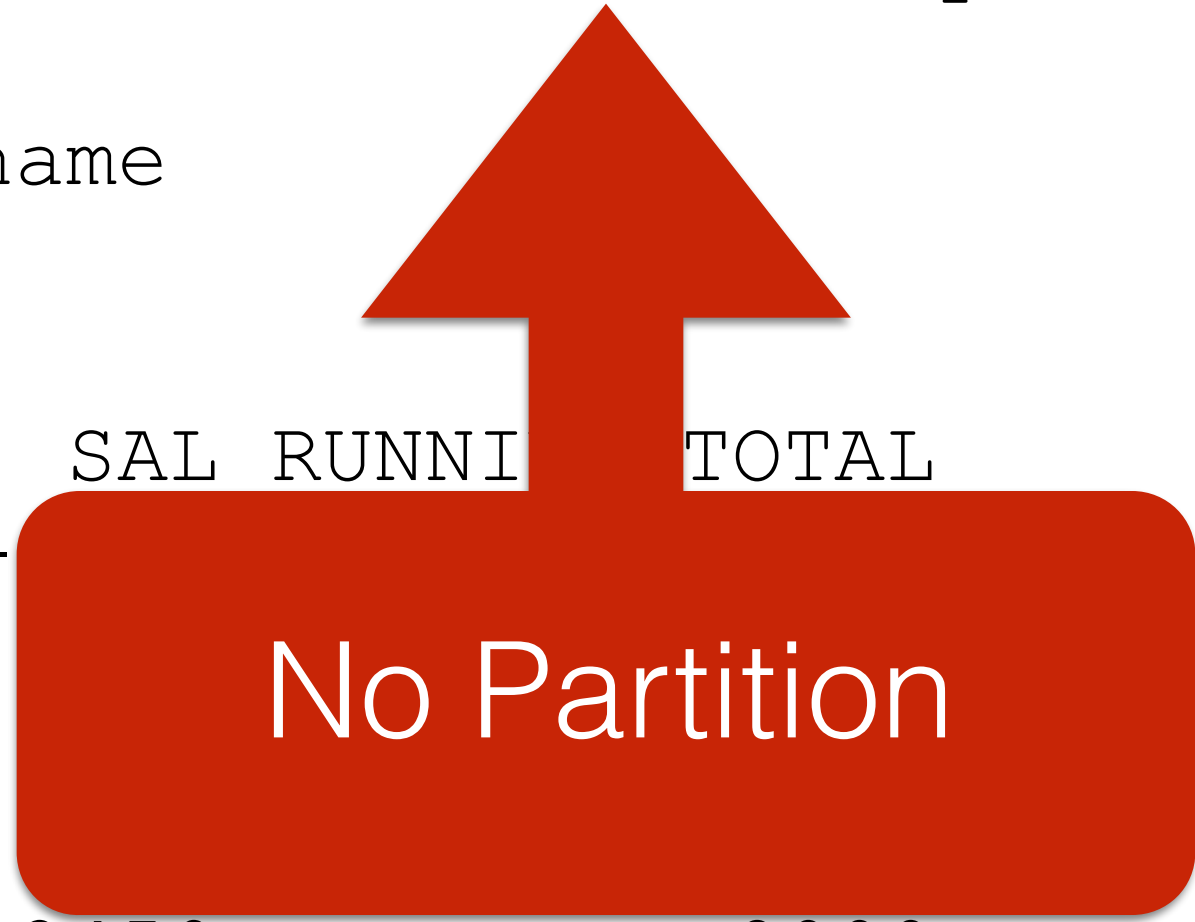
| ENAME  | SAL  | RUNNING_TOTAL |
|--------|------|---------------|
| ADAMS  | 1100 | 1100          |
| ALLEN  | 1600 | 2700          |
| BLAKE  | 2850 | 5550          |
| CLARK  | 2450 | 8000          |
| FORD   | 3000 | 11000         |
| JAMES  | 950  | 11950         |
| JONES  | 2975 | 14925     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

```
SQL> select 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 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  |      |               |
| ALLEN  |      |               |
| BLAKE  |      |               |
| 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         |

Default Window Clause:  
**Range Unbounded  
Preceding**

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

Running Total Per Department

```

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               |

```
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               |



```
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  | <b>1100</b> | <b>1100</b>        |
| 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               |

```

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  | <b>1100</b> | 1100               |
| 20     | FORD   | <b>3000</b> | <b>4100</b>        |
| 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               |

```

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  | <b>1100</b> | 1100               |
| 20     | FORD   | <b>3000</b> | 4100               |
| 20     | JONES  | <b>2975</b> | <b>7075</b>        |
| 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               |

Visualize the Window

```

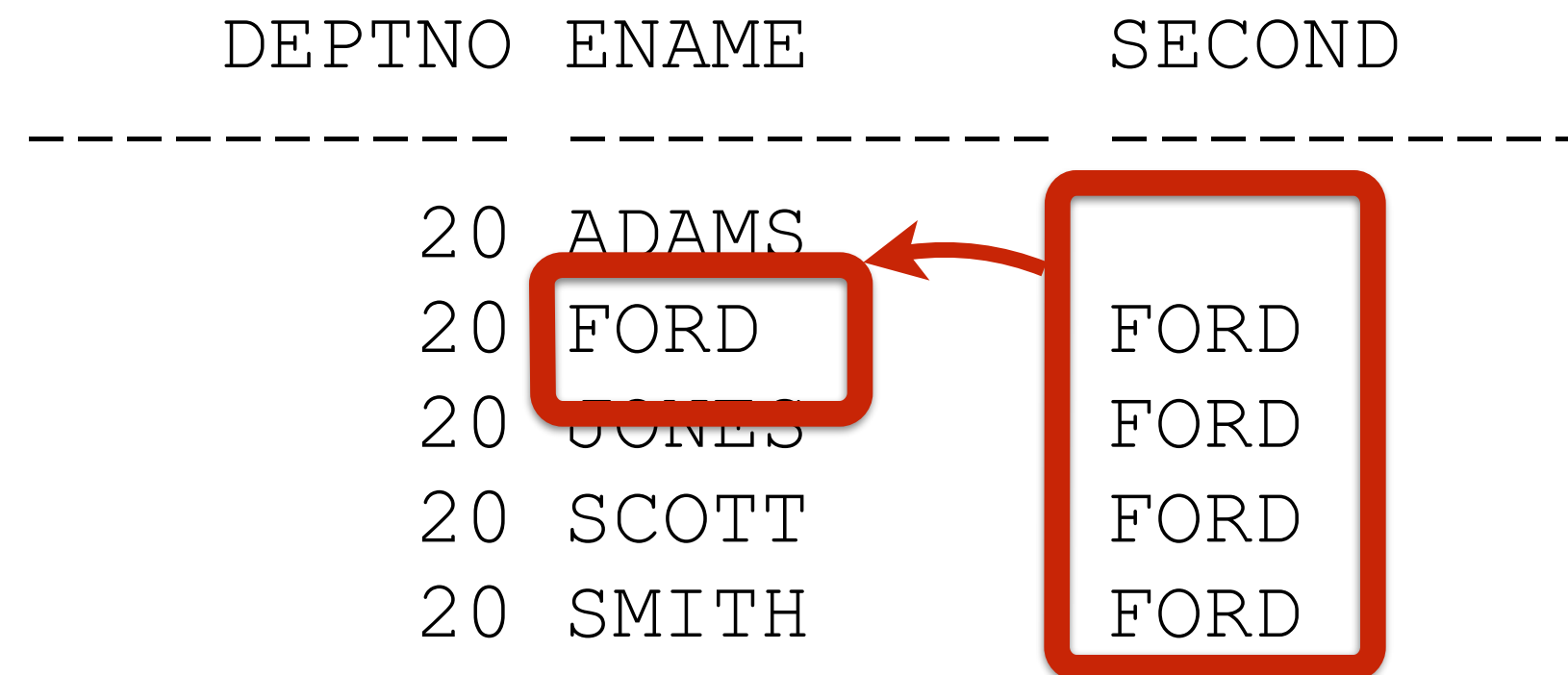
SQL> select deptno
2      , ename
3      , first_value (ename) over (partition by deptno
4                                     order by ename
5                                     ) fv
6      , last_value (ename) over (partition by deptno
7                                     order by ename
8                                     ) lv
9      from emp
10     where deptno = 20
11     ;

```

| DEPTNO | ENAME | FV    | LV    |
|--------|-------|-------|-------|
| 20     | ADAMS | ADAMS | ADAMS |
| 20     | FORD  | ADAMS | FORD  |
| 20     | JONES | ADAMS | JONES |
| 20     | SCOTT | ADAMS | SCOTT |
| 20     | SMITH | ADAMS | SMITH |

```
SQL> select deptno
2      , ename
3      , nth_value (ename, 2) from first
4          over (partition by deptno
5              order by ename
6              ) second
7      from emp
8      where deptno = 20
9      ;
```

| DEPTNO | ENAME | SECOND |
|--------|-------|--------|
| 20     | ADAMS |        |
| 20     | FORD  | FORD   |
| 20     | JONES | FORD   |
| 20     | SCOTT | FORD   |
| 20     | SMITH | FORD   |



```

SQL> select deptno
2      , ename
3      , nth_value (ename, 2) from last
4          over (partition by deptno
5                order by ename
6                ) second
7  from emp
8  where deptno = 20
9  ;

```

| DEPTNO | ENAME | SECOND |
|--------|-------|--------|
| 20     | ADAMS |        |
| 20     | FORD  | ADAMS  |
| 20     | JONES | FORD   |
| 20     | SCOTT | JONES  |
| 20     | SMITH | SCOTT  |

Ranking



## **ROW\_NUMBER**

Comparable to ROWNUM  
Arbitrarily

## **RANK**

Skip Values with Ties

## **DENSE\_RANK**

Doesn't Skip Values with Ties

```
SQL> select  ename
2           , sal
3           , row_number () over (order by sal desc) rn
4         from emp
5        where deptno = 20
6        order by sal desc
7        ;
```

| ENAME | SAL  | RN |
|-------|------|----|
| SCOTT | 3000 | 1  |
| FORD  | 3000 | 2  |
| JONES | 2975 | 3  |
| ADAMS | 1100 | 4  |
| SMITH | 800  | 5  |

```
SQL> select ename
2         , sal
3         , row_number () over (order by sal desc) rn
4     from emp
5     where deptno = 20
6     order by sal desc
7     ;
```

| ENAME | SAL  | RN |
|-------|------|----|
| SCOTT | 3000 | 1  |
| FORD  | 3000 | 2  |
| JONES | 2975 | 3  |
| ADAMS | 1100 | 4  |
| SMITH | 800  | 5  |

```
SQL> select ename
2         , sal
3         , row_number () over (order by sal desc) rn
4   from emp
5  where deptno = 20
6  order by sal desc
7  ;
```

| ENAME | SAL  | RN       |
|-------|------|----------|
| SCOTT | 3000 | <b>1</b> |
| FORD  | 3000 | <b>2</b> |
| JONES | 2975 | <b>3</b> |
| ADAMS | 1100 | <b>4</b> |
| SMITH | 800  | 5        |

```
SQL> select ename
2         , sal
3         , rank () over (order by sal desc) rk
4     from emp
5     where deptno = 20
6     order by sal desc
7     ;
```

| ENAME | SAL  | RK |
|-------|------|----|
| SCOTT | 3000 | 1  |
| FORD  | 3000 | 1  |
| JONES | 2975 | 3  |
| ADAMS | 1100 | 4  |
| SMITH | 800  | 5  |

```
SQL> select ename
2         , sal
3         , rank () over (order by sal desc) rk
4     from emp
5     where deptno = 20
6     order by sal desc
7     ;
```

| ENAME | SAL  | RK       |
|-------|------|----------|
| SCOTT | 3000 | <b>1</b> |
| FORD  | 3000 | <b>1</b> |
| JONES | 2975 | <b>3</b> |
| ADAMS | 1100 | <b>4</b> |
| SMITH | 800  | 5        |

```
SQL> select ename
2         , sal
3         , dense_rank () over (order by sal desc) dr
4     from emp
5     where deptno = 20
6     order by sal desc
7     ;
```

| ENAME | SAL  | DR |
|-------|------|----|
| SCOTT | 3000 | 1  |
| FORD  | 3000 | 1  |
| JONES | 2975 | 2  |
| ADAMS | 1100 | 3  |
| SMITH | 800  | 4  |

```
SQL> select ename
2         , sal
3         , dense_rank () over (order by sal desc) dr
4     from emp
5     where deptno = 20
6     order by sal desc
7     ;
```

| ENAME | SAL  | DR       |
|-------|------|----------|
| SCOTT | 3000 | <b>1</b> |
| FORD  | 3000 | <b>1</b> |
| JONES | 2975 | <b>2</b> |
| ADAMS | 1100 | <b>3</b> |
| SMITH | 800  | 4        |



```
SQL> select ename
2           ,sal
3           ,row_number() over (order by sal desc) rn
4           ,rank() over (order by sal desc) rk
5           ,dense_rank() over (order by sal desc) dr
6   from emp
7  where deptno = 20
8  order by sal desc
9  /
```

| ENAME | SAL  | RN       | RK       | DR       |
|-------|------|----------|----------|----------|
| SCOTT | 3000 | <b>1</b> | <b>1</b> | <b>1</b> |
| FORD  | 3000 | <b>2</b> | <b>1</b> | <b>1</b> |
| JONES | 2975 | <b>3</b> | <b>3</b> | <b>2</b> |
| ADAMS | 1100 | <b>4</b> | <b>4</b> | <b>3</b> |
| SMITH | 800  | 5        | 5        | 4        |

Top 3 per Department

```

SQL> select *
      2     from (select ename
      3                , deptno
      4                , sal
      5                , rank () over (partition by deptno
      6                                order by sal desc) rk
      7                from emp
      8                )
      9     where rk <= 3
     10     order by deptno
     11            , sal desc
     12 /

```

| ENAME  | DEPTNO | SAL   | RK    |
|--------|--------|-------|-------|
| -----  | -----  | ----- | ----- |
| KING   | 10     | 5000  | 1     |
| CLARK  | 10     | 2450  | 2     |
| MILLER | 10     | 1300  | 3     |
| SCOTT  | 20     | 3000  | 1     |
| FORD   | 20     | 3000  | 1     |
| JONES  | 20     | 2975  | 3     |
| BLAKE  | 30     | 2850  | 1     |
| ALLEN  | 30     | 1600  | 2     |
| TURNER | 30     | 1500  | 3     |

```

SQL> select *
  2     from (select ename
  3                , deptno
  4                , sal
  5                , rank () over (partition by deptno
  6                                order by sal desc) rk
  7     from emp
  8     )
  9     where rk <= 3
 10     order by deptno
 11            , sal desc
 12 /

```

| ENAME  | DEPTNO | SAL   | RK    |
|--------|--------|-------|-------|
| -----  | -----  | ----- | ----- |
| KING   | 10     | 5000  | 1     |
| CLARK  | 10     | 2450  | 2     |
| MILLER | 10     | 1300  | 3     |
| SCOTT  | 20     | 3000  | 1     |
| FORD   | 20     | 3000  | 1     |
| JONES  | 20     | 2975  | 3     |
| BLAKE  | 30     | 2850  | 1     |
| ALLEN  | 30     | 1600  | 2     |
| TURNER | 30     | 1500  | 3     |

```
SQL> select *
  2     from (select ename
  3                , deptno
  4                , sal
  5                , rank () over (partition by deptno
  6                                order by sal desc) rk
  7     from emp
  8     )
  9     where rk <= 3
 10     order by deptno
 11            , sal desc
 12 /
```

| ENAME  | DEPTNO | SAL  | RK |
|--------|--------|------|----|
| KING   | 10     | 5000 | 1  |
| CLARK  | 10     | 2450 | 2  |
| MILLER | 10     | 1300 | 3  |
| SCOTT  | 20     | 3000 | 1  |
| FORD   | 20     | 3000 | 1  |
| JONES  | 20     | 2975 | 3  |
| BLAKE  | 30     | 2850 | 1  |
| ALLEN  | 30     | 1600 | 2  |
| TURNER | 30     | 1500 | 3  |

PIVOT

```

SQL> select deptno
2      , max (decode (rn, 1, ename)) "Top 1"
3      , max (decode (rn, 2, ename)) "Top 2"
4      , max (decode (rn, 3, ename)) "Top 3"
5  from (select  ename
6              , deptno
7              , row_number() over (partition by deptno
8                                  order by sal desc
9                                  ) rn
10     from emp
11    )
12   where rn <= 3
13   group by deptno
14  ;

```

| DEPTNO | Top 1 | Top 2 | Top 3  |
|--------|-------|-------|--------|
| 10     | KING  | CLARK | MILLER |
| 20     | SCOTT | FORD  | JONES  |
| 30     | BLAKE | ALLEN | TURNER |

```

SQL> select deptno
2      , max (decode (rn, 1, ename)) "Top 1"
3      , max (decode (rn, 2, ename)) "Top 2"
4      , max (decode (rn, 3, ename)) "Top 3"
5      from (select ename
6                , deptno
7                , row_number() over (partition by deptno
8                                     order by sal desc
9                                     ) rn
10       from emp
11       )
12      where rn <= 3
13      group by deptno
14      ;

```

| DEPTNO | Top 1 | Top 2 | Top 3  |
|--------|-------|-------|--------|
| 10     | KING  | CLARK | MILLER |
| 20     | SCOTT | FORD  | JONES  |
| 30     | BLAKE | ALLEN | TURNER |



Real Pivot

```

SQL> select *
  2   from (select ename
  3             , deptno
  4             , row_number() over (partition by deptno
  5                                   order by sal desc
  6                                   ) rn
  7   from emp
  8   ) pivot (max (ename)
  9           for rn in (1,2,3) )
10  ;

```

| DEPTNO | 1     | 2     | 3      |
|--------|-------|-------|--------|
| 10     | KING  | CLARK | MILLER |
| 20     | SCOTT | FORD  | JONES  |
| 30     | BLAKE | ALLEN | TURNER |

```
SQL> select *
  2   from (select ename
  3             , deptno
  4             , row_number() over (partition by deptno
  5                                   order by sal desc
  6                                   ) rn
  7   from emp
  8   ) pivot (max (ename)
  9           for rn in (1,2,3) )
10 ;
```

| DEPTNO | 1     | 2     | 3      |
|--------|-------|-------|--------|
| 10     | KING  | CLARK | MILLER |
| 20     | SCOTT | FORD  | JONES  |
| 30     | BLAKE | ALLEN | TURNER |

Deduplication

```
SQL> select ename
2         , row_number() over (partition by ename
3                                 order by null
4                                 ) rn
5   from emp
6  where deptno = 20
7  ;
```

| ENAME | RN |
|-------|----|
|-------|----|

-----

|       |   |
|-------|---|
| ADAMS | 1 |
|-------|---|

|                  |              |
|------------------|--------------|
| <del>ADAMS</del> | <del>2</del> |
|------------------|--------------|

|      |   |
|------|---|
| FORD | 1 |
|------|---|

|      |   |
|------|---|
| FORD | 2 |
|------|---|

|       |   |
|-------|---|
| JONES | 1 |
|-------|---|

|                  |              |
|------------------|--------------|
| <del>JONES</del> | <del>2</del> |
|------------------|--------------|

|       |   |
|-------|---|
| SCOTT | 1 |
|-------|---|

|       |   |
|-------|---|
| SCOTT | 2 |
|-------|---|

|       |   |
|-------|---|
| SMITH | 1 |
|-------|---|

|                  |              |
|------------------|--------------|
| <del>SMITH</del> | <del>2</del> |
|------------------|--------------|



```
(select rid
  from (select rowid rid
        , row_number() over (partition by ename
                             order by null
                             ) rn
        from emp
        where deptno = 20
        )
 where rn > 1
)
```

```
SQL> delete from emp
2   where rowid in
3       (select rid
4          from (select rowid rid
5                  , row_number() over (partition by ename
6                                         order by null
7                                         ) rn
8          from emp
9          where deptno = 20
10         )
11        where rn > 1
12       )
13  /
```

5 rows deleted.



Stringing Together

| DEPTNO | ENAMES                                    |
|--------|---|
| 10     | CLARK, MILLER, KING                       |
| 20     | SMITH, FORD, ADAMS, SCOTT, JONES          |
| 30     | ALLEN, JAMES, TURNER, BLAKE, MARTIN, WARD |

```
SQL> create or replace type stragg_type as object
 2  (
 3    string varchar2(4000),
 4
 5    static function ODCIAggregateInitialize
 6      ( sctx in out stragg_type )
 7      return number ,
 8
 9    member function ODCIAggregateIterate
10      ( self in out stragg_type ,
11        value in      varchar2
12      ) return number ,
13
14    member function ODCIAggregateTerminate
15      ( self          in stragg_type,
16        returnvalue out varchar2,
17        flags in number
18      ) return number ,
19
20    member function ODCIAggregateMerge
21      ( self in out stragg_type,
22        ctx2 in      stragg_type
23      ) return number
24  )
```



```

SQL> create or replace type body stragg_type
 2  is
 3
 4  static function ODCIAggregateInitialize
 5  ( sctx in out stragg_type )
 6  return number
 7  is
 8  begin
 9
10      sctx := stragg_type( null ) ;
11
12      return ODCIConst.Success ;
13
14  end;
15
16  member function ODCIAggregateIterate
17  ( self in out stragg_type ,
18    value in      varchar2
19  ) return number
20  is
21  begin
22
23      self.string := self.string || ',' || value ;
24
25      return ODCIConst.Success;
26
27  end;
28
29  member function ODCIAggregateTerminate
30  ( self          in stragg_type ,
31    returnValue out varchar2 ,
32    flags        in number
33  ) return number
34  is
35  begin
36
37      returnValue := ltrim( self.string, ',' );
38
39      return ODCIConst.Success;
40
41  end;
42
43  member function ODCIAggregateMerge
44  ( self in out stragg_type ,
45    ctx2 in      stragg_type
46  ) return number
47  is
48  begin
49
50      self.string := self.string || ctx2.string;
51
52      return ODCIConst.Success;
53
54  end;
55
56  end;

```





```
SQL> create or replace function stragg
2     ( input varchar2 )
3     return varchar2
4     deterministic
5     parallel_enable
6     aggregate using stragg_type
7     ;
8     /
```

Function created.



```
SQL> select deptno
2         , stragg (ename) enames
3   from emp
4  group by deptno
5  ;
```

```
DEPTNO ENAMES
```

```
-----
```

|    |   |
|----|---|
| 10 | CLARK, MILLER, KING                       |
| 20 | SMITH, FORD, ADAMS, SCOTT, JONES          |
| 30 | ALLEN, JAMES, TURNER, BLAKE, MARTIN, WARD |



```
SQL> select deptno
2      , max (ltrim (sys_connect_by_path (ename, ','), ',')) enames
3    from (select ename
4              , deptno
5              , row_number() over (partition by deptno
6                                  order by sal
7                                  ) rn
8    from emp
9    )
10 start with rn = 1
11 connect by rn = prior rn + 1
12        and deptno = prior deptno
13 group by deptno
14 order by deptno
15 ;
```

DEPTNO ENAMES

---

```
10 MILLER, CLARK, KING
20 SMITH, ADAMS, JONES, SCOTT, FORD
30 JAMES, MARTIN, WARD, TURNER, ALLEN, BLAKE
```



```
SQL> select deptno
2         , rtrim (
3           xmlagg (
4             xmlelement (e, ename || ', ')).extract ('//text()')
5           , ', ') employees
6   from emp
7  group by deptno
8  /
```

DEPTNO EMPLOYEES

---

```
10 CLARK, MILLER, KING
20 SMITH, FORD, ADAMS, SCOTT, JONES
30 ALLEN, JAMES, TURNER, BLAKE, MARTIN, WARD
```



```
SQL> create type big_varchar2  
2 as table of varchar2(4000)  
3 ;  
4 /
```

Type created.

```
SQL> select deptno
2      , cast (collect (ename order by sal) as big_varchar2) enames
3    from emp
4   group by deptno
5   ;
```

```
DEPTNO ENAMES
```

```
-----
10 BIG_VARCHAR2 ('MILLER', 'CLARK', 'KING')
20 BIG_VARCHAR2 ('SMITH', 'ADAMS', 'JONES', 'SCOTT', 'FORD')
30 BIG_VARCHAR2 ('JAMES', 'WARD', 'MARTIN', 'TURNER', 'ALLEN', '
   BLAKE')
```

# Undocumented

```
SQL> select deptno
2      , wm_concat (ename) enames
3    from emp
4   group by deptno
5  ;
```

```
DEPTNO ENAMES
```

---

```
10 CLARK, MILLER, KING
20 SMITH, FORD, ADAMS, SCOTT, JONES
30 ALLEN, JAMES, TURNER, BLAKE, MARTIN, WARD
```

« [SQL\\*Plus error loggi...](#) | [Main](#)

## Why not use WM\_CONCAT function in Oracle?

By LalitKumarB-Oracle on Jun 11, 2015

String aggregation related questions are frequently asked questions and every now and then someone provides a solution using **WM\_CONCAT** function. And it really annoys me to repeat the same thing every time.

Any application which has had been relying on **WM\_CONCAT** function will not work once upgraded to **12c**. Since, it has been removed from the latest 12c version.

```
SQL> select banner from v$version where rownum = 1;
```

```
BANNER
```

```
-----  
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
```

```
SQL> SELECT object_name  
2 FROM dba_objects  
3 WHERE owner='WMSYS'  
4 AND object_name LIKE 'WM\_%' ESCAPE '\';
```

### About

My blog is dedicated to Oracle Database Application Development and Database Administration. I always wondered why people do the blogging thing. Especially, for technical stuff, we already have the fine manuals, documentations, standard operating procedures etc., then, what's the big deal in posting technical stuff in a blog? Well, I got my answer when I actually made my first post. A good blog post is about gathering rich information based on our own experience. The test cases that we prepare, are a result of the experience and expertise in that field. We often learn things from various sources like books, online docs, videos,

« [SQL\\*Plus error loggi...](#) | [Main](#)

## About

My blog is dedicated to Oracle Database Application Development and Database Administration. I always wondered why people do the blogging thing. Especially, for

ave the  
ns,

standard operating procedures etc., then, what's the big deal in posting technical stuff in a blog? Well, I got my answer when I actually made my first post. A good blog post is about gathering rich information based on our own experience. The test cases that we prepare, are a result of the experience and expertise in that field. We often learn things from various sources like books, online docs, videos,

## Why not use WM\_CONCAT function in Oracle?

By LalitKumarB-Oracle on Jun 11, 2015

String aggregation related questions are frequently asked questions and every now and then someone provides a solution.

Any application which has had been relying on **WM\_CONCAT** function will not work once upgraded to **12c**. Since, it has been removed from the latest 12c version.

Any application which has had been relying on **WM\_CONCAT** function will not work once upgraded to **12c**. Since, it has been removed from the latest 12c version.

```
SQL> select banner from v$version where rownum = 1;
```

BANNER

```
-----  
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
```

```
SQL> SELECT object_name  
2 FROM dba_objects  
3 WHERE owner='WMSYS'  
4 AND object_name LIKE 'WM\_%' ESCAPE '\';
```

```
SQL> select deptno
2         , listagg( ename, ',' ) within group (order by sal)
3         enames
4     from emp
5     group by deptno
6     ;
```

DEPTNO ENAMES

---

10 MILLER, CLARK, KING

20 SMITH, ADAMS, JONES, FORD, SCOTT

30 JAMES, MARTIN, WARD, TURNER, ALLEN, BLAKE

```
SQL> select deptno
2      , listagg( ename, ',' ) within group (order by sal)
3          over (partition by deptno)
4          enames
5      from emp
6     where deptno = 20
7     ;
```

DEPTNO ENAMES

---

```
20 SMITH,ADAMS,JONES,FORD,SCOTT
20 SMITH,ADAMS,JONES,FORD,SCOTT
20 SMITH,ADAMS,JONES,FORD,SCOTT
20 SMITH,ADAMS,JONES,FORD,SCOTT
20 SMITH,ADAMS,JONES,FORD,SCOTT
```

Access to Other Rows



```
SQL> select ename
2      , lead (ename) over (partition by deptno
3                                order by ename
4                                )
5      from emp
6      where deptno = 20
7      ;
```

| ENAME | LEAD (ENAME |
|-------|-------------|
| ADAMS | FORD        |
| FORD  | JONES       |
| JONES | SCOTT       |
| SCOTT | SMITH       |
| SMITH |             |

The diagram illustrates the result of the LEAD function. The 'LEAD (ENAME' column is shifted one row up relative to the 'ENAME' column. Red boxes highlight 'FORD' in both columns, with an arrow pointing from the 'LEAD (ENAME' 'FORD' to the 'ENAME' 'FORD'.

```
SQL> select ename
2           , lead (ename, 2) over (partition by deptno
3                                     order by ename
4                                     )
5       from emp
6       where deptno = 20
7       ;
```

| ENAME | LEAD (ENAME |
|-------|-------------|
| ADAMS | JONES       |
| FORD  | SCOTT       |
| JONES | SMITH       |
| SCOTT |             |
| SMITH |             |

```
SQL> select ename
2         , lead (ename, 2, ' - Outside - ')
3         over (partition by deptno
4              order by ename
5              )
6   from emp
7  where deptno = 20
8  ;
```

| ENAME | LEAD (ENAME, 2, |
|-------|-----------------|
| ----- | -----           |
| ADAMS | JONES           |
| FORD  | SCOTT           |
| JONES | SMITH           |
| SCOTT | - Outside -     |
| SMITH | - Outside -     |

Next NOT NULL

| ENAME  | COMM | NEXT_COMM |
|--------|------|-----------|
| ALLEN  | 300  | 1400      |
| BLAKE  |      | 1400      |
| JAMES  |      | 1400      |
| MARTIN | 1400 | 0         |
| TURNER | 0    | 500       |
| WARD   | 500  |           |

```
SQL> select ename
2           , comm
3           , lead (comm) over (partition by deptno
4                                   order by ename
5                                   ) next_comm
6   from emp
7  where deptno = 30
8  order by ename
9  ;
```

| ENAME  | COMM | NEXT_COMM |
|--------|------|-----------|
| ALLEN  | 300  |           |
| BLAKE  |      |           |
| JAMES  |      | 1400      |
| MARTIN | 1400 | 0         |
| TURNER | 0    | 500       |
| WARD   | 500  |           |

```

SQL> select  ename
2           , comm
3           , max(comm) over (order by grp
4                               range between 1 following
5                               and 1 following
6                               ) next_comm
7   from ( select  ename
8             , comm
9             , sum(nvl2(comm,1,0)) over (order by ename) grp
10          from emp
11          where deptno = 30
12          )
13   order by ename
14   ;

```

| ENAME  | COMM | NEXT_COMM |
|--------|------|-----------|
| ALLEN  | 300  | 1400      |
| BLAKE  |      | 1400      |
| JAMES  |      | 1400      |
| MARTIN | 1400 | 0         |
| TURNER | 0    | 500       |
| WARD   | 500  |           |

```
SQL> select ename
2          , comm
3          , lead (comm) ignore nulls
4          over (partition by deptno
5                order by ename
6                ) next_comm
7  from emp
8  where deptno = 30
9  order by ename
10 ;
```

| ENAME  | COMM | NEXT_COMM |
|--------|------|-----------|
| ALLEN  | 300  | 1400      |
| BLAKE  |      | 1400      |
| JAMES  |      | 1400      |
| MARTIN | 1400 | 0         |
| TURNER | 0    | 500       |
| WARD   | 500  |           |



Looking Back

```

SQL> select ename
2          , lag (ename) over (partition by deptno
3                                     order by ename
4                                     )
5      from emp
6     where deptno = 20
7     ;

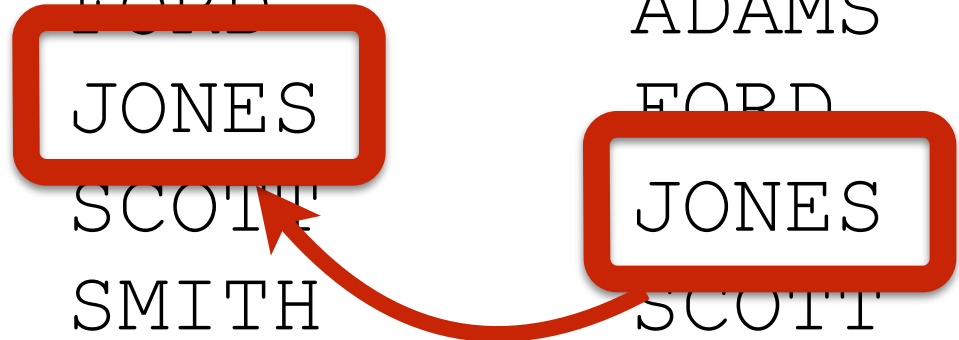
```

| ENAME | LAG (ENAME) |
|-------|-------------|
| ADAMS |             |
| FORD  | ADAMS       |
| JONES | FORD        |
| SCOTT | JONES       |
| SMITH | SCOTT       |

```

lag (ename, 2)
lag (ename, 2, '-outside-')

```



Rounding Last Record

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

| ID |         |
|----|---------|
| 43 | \$33.33 |
| 44 | \$33.33 |
| 45 | \$33.33 |

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

| ID |                |
|----|----------------|
| 43 | \$33.33        |
| 44 | \$33.33        |
| 45 | <b>\$33.34</b> |

```
SQL> select id
2         ,lead (null, 1, 'x') over (order by id) lastrow
3   from t
4   /
```

| ID | L |
|----|---|
| 43 | - |
| 44 | - |
| 45 | x |

```

SQL> select id
2         ,rounded +
3         case
4           when lastrow = 'x'
5             then amount - sum (rounded) over ()
6             else 0
7           end as final_amount
8   from (
9     select id
10          ,100 amount
11          ,round (100 / count(*) over (), 2) rounded
12          ,lead (null, 1, 'x') over (order by id) lastrow
13     from t
14   )
15 /

```

| ID | FINAL_AMOUNT |
|----|--------------|
| 43 | 33.33        |
| 44 | 33.33        |
| 45 | 33.34        |

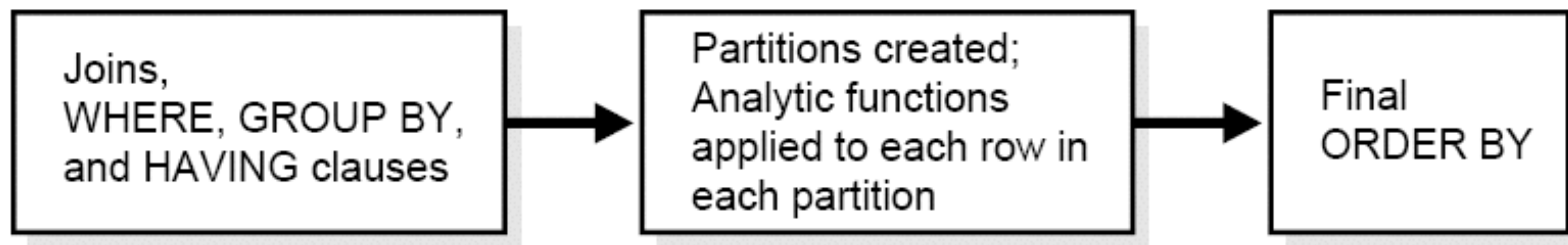
Caveats



Cannot be in Final Predicate

Order of NULL

Possible Performance Impact



A close-up photograph of a festive dinner table. In the center, a dark, round pudding sits on a silver platter with a blue inner ring. A sprig of holly is stuck into the top of the pudding. To the right, a bottle of Henley Martin is visible. In the background, there is a white teacup with floral patterns on a saucer, a golden chalice, and a bookshelf. The scene is lit with warm, low-key lighting, creating a cozy atmosphere.

Proof of the Pudding



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 future
- Highest Start Date, nulls are ignored
- Highest ID
- *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))
)
```









→ *per Category*

→ *per Category*  
partition by

Which Analytic Function?





A glowing lightbulb is centered in the background, emitting a soft, warm light. The bulb is slightly out of focus, creating a bokeh effect. The light from the bulb illuminates the surrounding area, creating a gradient of light colors. The text "Rephrase the Requirement" is overlaid on the lightbulb, appearing in a white, sans-serif font. The text is centered horizontally and vertically, with the lightbulb's glow providing a subtle background for the words.

Rephrase the Requirement



"Most Recent Contract"



"Top One Contract"



- Highest End Date, nulls are future
- Highest Start Date, nulls are ignored
- Highest ID
- per Category

- Highest End Date, nulls are future
- Highest Start Date, nulls are ignored
- Highest ID
- **per Category**

partition by cat

→ **Highest End Date, nulls are future**

→ Highest Start Date, nulls are ignored

→ Highest ID

→ per Category

```
partition by cat  
order by end_date desc nulls first
```

- Highest End Date, nulls are future
- **Highest Start Date, nulls are ignored**
- Highest ID
- per Category

```
partition by cat
  order by end_date desc nulls first
         , start_date desc nulls last
```

- Highest End Date, nulls are future
- Highest Start Date, nulls are ignored
- **Highest ID**
- per Category

```
partition by cat
  order by end_date desc nulls first
         , start_date desc nulls last
         , id desc
```

```
, row_number() over (  
    partition by cat  
    order by end_date desc nulls first  
           , start_date desc nulls last  
           , id desc  
    ) rn
```



```
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
```

```
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
```

Original Query: 00:00:00.01

Analytic Function: 00:00:00.03

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                         |

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

---

|       |  |
|-------|--|
| 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                         |

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    | 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                         |

| Records | Original Query | Analytical Function |
|---------|----------------|---------------------|
|---------|----------------|---------------------|

60

00:00:00.01

00:00:00.03

|      |             |             |
|------|-------------|-------------|
| 7680 | 00:00:01.65 | 00:00:00.07 |
|------|-------------|-------------|

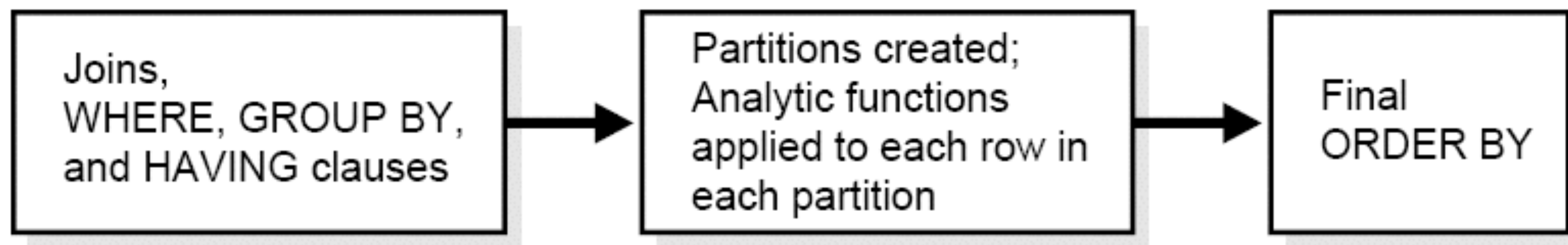
1966080

> 3 hours

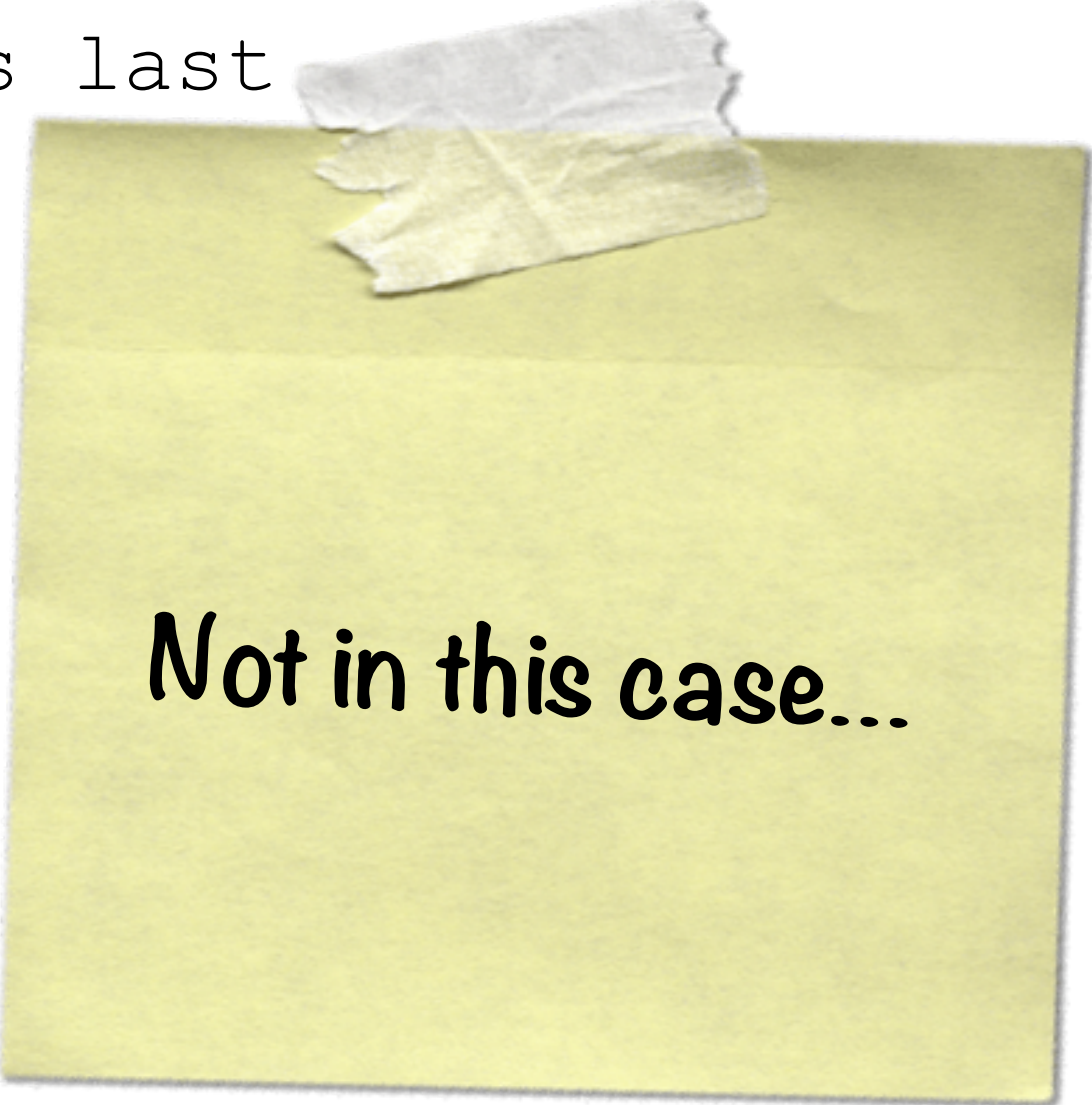
00:00:08.37

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





```
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...**

Take a step back and evaluate other options

There are multiple ways to solve a requirement

Test with a Representative Set of Data

Comment your Code!

"Best thing since SELECT statement"

*Tom Kyte*

Alex Nuijten  
Alex.Nuijten@Ordina.nl

nuijten.blogspot.com  
@alexnuijten



Q & A



Alex Nuijten

**ORDINA**



*photosteve101*

*<https://flic.kr/p/bnZKrV>*



*<http://www.twing.nl/images/dagtip/amcharts.gif>*



*<http://ad.nl>*



*James Petts*

*<https://flic.kr/p/qwTT4R>*

