



ORACLE ANALYTICAL FUNCTIONS

Vijay Mahawar

<https://whizdba.wordpress.com/>

Saturday, 18 February, 2012

ABOUT ME



I am **Vijay Mahawar**, an Oracle Technologist. I am a member of *AIOUG*, *ODTUG* and *OTN*. I am certified in Oracle and hold OCP in Oracle 11g DBA and OCP in Oracle 11g Advanced Developer Track. I am also OPN Certified Specialist.



I am associated with Oracle Technologies since last 10 years. During this time I have held positions of Trainer, Developer, DBA, Analyst and Consultant in Companies like L&T Infotech Ltd., HCL Technologies Ltd., Dell Services and Sapient Consulting Pvt. Ltd.

TARGET AUDIENCE

- SQL Developers
- PL/SQL Developers
- DBAs
- Database Architects

AGENDA

- Introduction to Analytical Functions
- Analytical Functions vs. Group or Aggregate functions
- LEAD and LAG
- FIRST VALUE and LAST VALUE
- FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

INTRODUCTION TO ANALYTICAL FUNCTIONS

- Built on Aggregate Functions
- Row-wise Group Data
- Complex Operations (Statistical, Reporting, etc.)
- NOT a replacement for Aggregate Functions

- [Introduction to Analytical Functions](#)
- Analytical Functions vs. Aggregate functions
- LEAD and LAG
- FIRST VALUE and LAST VALUE
- FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

- ✓ Introduction to Analytical Functions
- [Analytical Functions vs. Aggregate functions](#)
- LEAD and LAG
- FIRST VALUE and LAST VALUE
- FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

ANALYTICAL FUNCTIONS LIST

Which depends on order of records

- LEAD
- LAG
- RANK
- DENSE_RANK
- ROW_NUMBER
- FIRST
- FIRST VALUE
- LAST
- LAST VALUE

Which does not depend on order of records

- SUM
- COUNT
- AVG
- MIN
- MAX

Newly introduced in Oracle 11gR2

LISTAGG
NTH_VALUE

- ✓ Introduction to Analytical Functions
- [Analytical Functions vs. Aggregate functions](#)
- LEAD and LAG
- FIRST VALUE and LAST VALUE
- FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

ANALYTICAL FUNCTIONS LIST

Depends on order of records

- LEAD
- LAG
- RANK
- DENSE_RANK
- ROW_NUMBER
- FIRST
- FIRST VALUE
- LAST
- LAST VALUE

Do not depend on order of records

- SUM
- COUNT
- AVG
- MIN
- MAX



Aggregate Functions

Newly introduced in Oracle 11gR2
LISTAGG
NTH_VALUE

- ✓ Introduction to Analytical Functions
- [Analytical Functions vs. Aggregate functions](#)
- LEAD and LAG
- FIRST VALUE and LAST VALUE
- FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

ANALYTICAL FUNCTIONS LIST

○ Syntax:

*Function(arg1,..., argn) OVER ([PARTITION BY <...>]
[ORDER BY <...>] [<window_clause>])*

○ Example:

*select deptno, empno, ename, **Count(empno) OVER (PARTITION
BY deptno ORDER BY ename)** as cnt_eno from emp;*

- ✓ Introduction to Analytical Functions
- [Analytical Functions vs. Aggregate functions](#)
- LEAD and LAG
- FIRST VALUE and LAST VALUE
- FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

ANALYTICAL FUNCTIONS LIST

○ Syntax:

*Function(arg1,..., argn) OVER ([PARTITION BY <...>]
[ORDER BY <...>] [<window_clause>])*

○ Example:

*select deptno, empno, ename, **Count(empno) OVER (PARTITION
BY deptno ORDER BY ename)** as cnt_eno from emp;*

10	7782	CLARK	1
10	7839	KING	2
10	7934	MILLER	3
20	7876	ADAMS	1
20	7902	FORD	2
20	7566	JONES	3
20	7788	SCOTT	4
20	7369	SMITH	5
20	7499	NEWMAN	6

- ✓ Introduction to Analytical Functions
- [Analytical Functions vs. Aggregate functions](#)
- LEAD and LAG
- FIRST VALUE and LAST VALUE
- FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

ANALYTICAL FUNCTIONS LIST

○ Syntax:

*Function(arg1,..., argn) OVER ([PARTITION BY <...>]
[ORDER BY <...>] [<window_clause>])*

○ Example:

*select deptno, empno, ename, **Count(empno) OVER (PARTITION BY deptno ORDER BY ename)** as cnt_eno from emp;*

```
SQL> select count(empno) from emp;

COUNT (EMPNO)
-----
                14
```

```
10      7782 CLARK      1
10      7839 KING       2
10      7934 MILLER    3
20      7876 ADAMS      1
20      7902 FORD      2
20      7566 JONES     3
20      7788 SCOTT     4
20      7369 SMITH     5
20      7499 ...       6
```

- ✓ Introduction to Analytical Functions
- [Analytical Functions vs. Aggregate functions](#)
- LEAD and LAG
- FIRST VALUE and LAST VALUE
- FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

ANALYTICAL FUNCTIONS LIST

○ Syntax:

*Function(arg1,..., argn) OVER ([PARTITION BY <...>]
[ORDER BY <...>] [<window_clause>])*

○ Example:

*select deptno, empno, ename, **Count(empno) OVER (PARTITION BY deptno ORDER BY ename)** as cnt_eno from emp;*

```
SQL> select count(empno) from emp;

COUNT (EMPNO)
-----
              14
```

Aggregate Function

```
10      7782 CLARK      1
10      7839 KING      2
10      7934 MILLER   3
20      7876 ADAMS    1
20      7902 FORD     2
20      7566 JONES    3
20      7388 SCOTT    4
        7369 SMITH     5
        7499 ...
```

Analytical Function

- ✓ Introduction to Analytical Functions
- ✓ Analytical Functions vs. Aggregate functions
- **LEAD and LAG**
- FIRST VALUE and LAST VALUE
- FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

LEAD AND LAG

- Computes an expression on the next rows and return the value to the current row.

LEAD (<sql_expr>, <offset>, <default>) OVER (<analytic_clause>)

<sql_expr> expression to compute from the leading row.

<offset> index of the leading row relative to the current row.

<offset> positive integer with default 1.

<default> is the value to return if the <offset> points to a row outside the partition range.

- The syntax of LAG is similar but offset goes into the previous rows.

- ✓ Introduction to Analytical Functions
- ✓ Analytical Functions vs. Aggregate functions
- ✓ LEAD and LAG
- **FIRST VALUE and LAST VALUE**
- FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

FIRST VALUE AND LAST VALUE

- The `FIRST_VALUE` analytic function picks the first record from the partition after doing the `ORDER BY`.

- Syntax:

`FIRST_VALUE(<sql_expr>) OVER (<analytic_clause>)`

The `<sql_expr>` is computed on the columns of this first record and results are returned.

- The `LAST_VALUE` function is used in similar context except that it acts on the last record of the partition.

- ✓ Introduction to Analytical Functions
- ✓ Analytical Functions vs. Aggregate functions
- ✓ LEAD and LAG
- ✓ FIRST VALUE and LAST VALUE
- **FIRST and LAST**
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

FIRST AND LAST

- The FIRST function also called KEEP FIRST function
- Suppose we rank a group of record and found several records in the first rank. Now we want to apply an aggregate function on the records of the first rank. KEEP FIRST enables that.

- Syntax:

Function() KEEP (DENSE_RANK FIRST ORDER BY <expr>) OVER (<partitioning_clause>)

- FIRST and LAST functions syntax deviates from the general syntax
- Don't have ORDER BY
- Don't support <window> clause.
- FIRST and LAST always use DENSE_RANK

- ✓ Introduction to Analytical Functions
- ✓ Analytical Functions vs. Aggregate functions
- ✓ LEAD and LAG
- ✓ FIRST VALUE and LAST VALUE
- ✓ FIRST and LAST
- [Window clause \(ROW and RANGE\)](#)
- LISTAGG and NTH_VALUE - **new in 11gR2**
- Performance tips

WINDOW CLAUSE

- Analytic functions which take a window clause
 - AVG
 - COUNT
 - FIRST_VALUE
 - LAST_VALUE
 - MAX
 - MIN
 - SUM
- Important feature:
DYNAMIC IN NATURE

- ✓ Introduction to Analytical Functions
- ✓ Analytical Functions vs. Aggregate functions
- ✓ LEAD and LAG
- ✓ FIRST VALUE and LAST VALUE
- ✓ FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - new in 11gR2
- Performance tips

WINDOW CLAUSE

- Syntax:
[ROW or RANGE] BETWEEN <start_expr> AND <end_expr>
- <start_expr> can be any one of the following
 - UNBOUNDED PRECEDING
 - CURRENT ROW
 - <sql_expr> PRECEDING or FOLLOWING.
- <end_expr> can be any one of the following
 - UNBOUNDED FOLLOWING or
 - CURRENT ROW or
 - <sql_expr> PRECEDING or FOLLOWING.

- ✓ Introduction to Analytical Functions
- ✓ Analytical Functions vs. Aggregate functions
- ✓ LEAD and LAG
- ✓ FIRST VALUE and LAST VALUE
- ✓ FIRST and LAST
- Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - new in 11gR2
- Performance tips

WINDOW CLAUSE

- For ROW type windows the definition is in terms of row numbers before or after the current row. So for ROW type windows `<sql_expr>` must evaluate to a positive integer.
- For RANGE type windows the definition is in terms of values before or after the current ORDER.
- The ROW or RANGE window cannot appear together in one OVER clause. The window clause is defined in terms of the current row. But may or may not include the current row. The start point of the window and the end point of the window can finish before the current row or after the current row.
- Only start point cannot come after the end point of the window. In case any point of the window is undefined the default is UNBOUNDED PRECEDING for `<start_expr>` and UNBOUNDED FOLLOWING for `<end_expr>`.

- ✓ Introduction to Analytical Functions
- ✓ Analytical Functions vs. Aggregate functions
- ✓ LEAD and LAG
- ✓ FIRST VALUE and LAST VALUE
- ✓ FIRST and LAST
- ✓ Window clause (ROW and RANGE)
- [LISTAGG and NTH_VALUE - new in 11gR2](#)
- Performance tips

LISTAGG AND NTH_VALUE

- The LISTAGG analytic function was introduced in Oracle 11g Release 2, making it very easy to aggregate strings.
- The nice thing about this function is it also allows us to order the elements in the concatenated list.
- Example:

```
SELECT deptno,  
LISTAGG(ename, ',') WITHIN GROUP  
(ORDER BY ename) AS employees  
FROM emp  
GROUP BY deptno;
```

- ✓ Introduction to Analytical Functions
- ✓ Analytical Functions vs. Aggregate functions
- ✓ LEAD and LAG
- ✓ FIRST VALUE and LAST VALUE
- ✓ FIRST and LAST
- ✓ Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - new in 11gR2
- Performance tips

LISTAGG AND NTH_VALUE

- The LISTAGG analytic function was introduced in Oracle 11g Release 2, making it very easy to aggregate strings.
- The nice thing about this function is it also allows us to order the elements in the concatenated list.
- Example for LISTAGG is as follows:

```
SELECT deptno,  
LISTAGG(ename, ',') WITHIN GROUP  
(ORDER BY ename) AS employees  
FROM emp  
GROUP BY deptno;
```

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

3 rows selected.
```

- ✓ Introduction to Analytical Functions
- ✓ Analytical Functions vs. Aggregate functions
- ✓ LEAD and LAG
- ✓ FIRST VALUE and LAST VALUE
- ✓ FIRST and LAST
- ✓ Window clause (ROW and RANGE)
- [LISTAGG and NTH_VALUE - new in 11gR2](#)
- Performance tips

LISTAGG AND NTH_VALUE

- Introduced in 11gR2.
- Unlike FIRST_VALUE and LAST_VALUE which are used to find the first respectively last value in a window in a partition that has been ordered in a certain way.
- Analytical Function NTH_VALUE returns the NTH_VALUE in a WINDOW

FIRST_VALUE(column) == NTH_VALUE(column,1)

LAST_VALUE(column) == NTH_VALUE(column,1) FROM LAST

- Example:

select deptno ,

ename ,

first_value(sal) over (partition by deptno order by sal desc) -

*nth_value(sal,2) from first over (partition by deptno order by sal
 desc) top2_difference*

from emp

- ✓ Introduction to Analytical Functions
- ✓ Analytical Functions vs. Aggregate functions
- ✓ LEAD and LAG
- ✓ FIRST VALUE and LAST VALUE
- ✓ FIRST and LAST
- ✓ Window clause (ROW and RANGE)
- LISTAGG and NTH_VALUE - new in 11gR2
- Performance tips

LISTAGG AND NTH_VALUE

- Introduced in 11gR2.
- Unlike FIRST_VALUE and LAST_VALUE which are used to find the first respectively last value in a window in a partition that has been ordered in a certain way.

- Analytical Function NTH_VALUE returns the NTH_VALUE FROM FIRST WINDOW

FIRST_VALUE(column) == NTH_VALUE(column,1)

LAST_VALUE(column) == NTH_VALUE(column,1) FROM FIRST WINDOW

- Example of NTH_VALUE is:

select deptno ,

ename ,

first_value(sal) over (partition by deptno order by sal desc) -

nth_value(sal,2) from first over (partition by deptno order by sal desc) top2_difference

from emp

DEPTNO	ENAME	TOP2_DIFFERENCE
10	KING	
10	CLARK	2550
10	MILLER	2550
20	SCOTT	0
20	FORD	0
20	JONES	0
20	ADAMS	0
20	SMITH	0
30	BLAKE	
30	ALLEN	1250
30	TURNER	1250
30	MARTIN	1250
30	WARD	1250
30	JAMES	1250

- ✓ Introduction to Analytical Functions
- ✓ Analytical Functions vs. Aggregate functions
- ✓ LEAD and LAG
- ✓ FIRST VALUE and LAST VALUE
- ✓ FIRST and LAST
- ✓ Window clause (ROW and RANGE)
- ✓ LISTAGG and NTH_VALUE - new in 11gR2
- Performance tips

PERFORMANCE TIPS

- Defining the PARTITION BY and ORDER BY clauses on indexed columns
- Use CBO for queries using analytic functions.
- The tables and indexes should be analyzed.
- Optimizer mode should be CHOOSE.
- Sorting for computing partition and order by clause takes time.
- Queries with multiple analytic functions,
- Sorting and Partitioning on two different non-indexed columns should be avoided in same query.

QUESTIONS?

SOME USEFUL LINKS

http://docs.oracle.com/cd/E11882_01/server.112/e26088/functions004.htm

<http://www.oracle-base.com/articles/misc/StringAggregationTechniques.php#listagg>

http://psoug.org/reference/analytic_functions.html

My Blog : <https://whizdba.wordpress.com>



vmahawar (on twitter)

<https://whizdba.wordpress.com>

THANK YOU