

# Advanced SQL in Oracle and SQL Server

## Analytic Functions – Part III

Scott L. Hecht

<http://www.sheepsqueezers.com>

@sheepsqueezers



**pluralsight**  
hardcore developer training



# Module Contents

- **Analytic Functions**
  - Data used in Module
  - Windowing (ROWS/RANGE) Clause
  - Summary

# Data Used in Module

- **Table**

- CHILDSTAT

- **Columns**

- FIRSTNAME – child's first name
  - GENDER – child's gender (M=Male, F=Female)
  - BIRTHDATE – child's date of birth
  - HEIGHT – child's height (inches)
  - WEIGHT – child's weight (pounds)

- **Data**

<u>FIRSTNAME</u>	<u>GENDER</u>	<u>BIRTHDATE</u>	<u>HEIGHT</u>	<u>WEIGHT</u>
LAUREN	F	10-JUN-00	54	876
ROSEMARY	F	08-MAY-00	35	123
ALBERT	M	02-AUG-00	45	150
BUDDY	M	02-OCT-98	45	189
FARQUAR	M	05-NOV-98	76	198
SIMON	M	03-JAN-99	87	256
TOMMY	M	11-DEC-98	78	167



# Window Clause

- **What is the Window Clause?**
  - Using Analytic Functions allows us to access data very different from the normal sequential way!
  - PARTITION BY chunks up data
  - LEAD()/LAG() allow access to a particular row before or after current row
  - FIRST\_VALUE()/LAST\_VALUE() allow access to first/last row
- **Gain more control as we move down this list!**
- **The Window Clause allows even more fine-grained access!**
  - No WINDOW keyword! Use ROWS or RANGE instead.
  - If no Window Clause specified, default is current row on back.
  - Availability:
    - Oracle: 8i
    - SQL Server: 2012
- **Default Syntax:**

**ROWS BETWEEN UNBOUNDED PRECEDING  
AND CURRENT ROW**

## Example #14



- Task: Re-do previous example (Retrieve the name of the heaviest and lightest male and female child.)
- Note: Include *default* Window Clause syntax!

```
SELECT A.FIRSTNAME, A.GENDER, A.WEIGHT,
       FIRST_VALUE(A.FIRSTNAME) OVER (PARTITION BY A.GENDER
                                       ORDER BY A.WEIGHT
                                       ROWS BETWEEN UNBOUNDED PRECEDING
                                              AND CURRENT ROW)
                                              AS LT_CHILD,
       LAST_VALUE(A.FIRSTNAME) OVER (PARTITION BY A.GENDER
                                       ORDER BY A.WEIGHT
                                       ROWS BETWEEN UNBOUNDED PRECEDING
                                              AND CURRENT ROW)
                                              AS HV_CHILD
FROM CHILDSTAT A
ORDER BY A.GENDER, A.WEIGHT
```

## Example #14

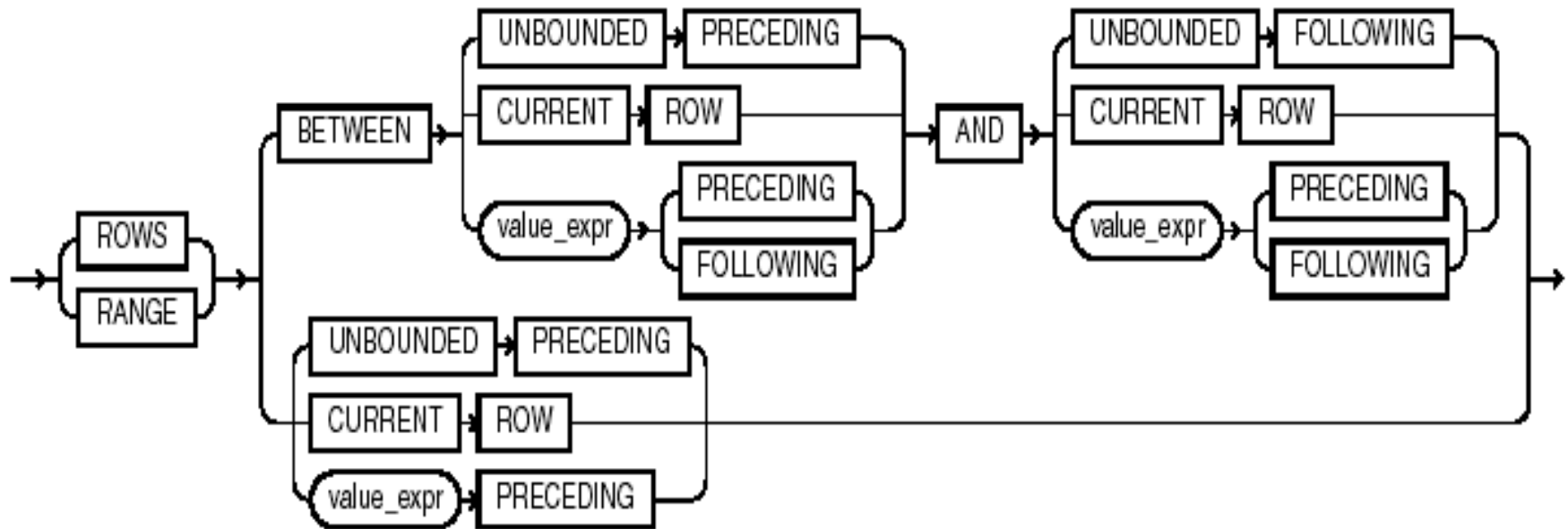


- Task: Re-do previous example (Retrieve the name of the heaviest and lightest male and female child.)
- Note: Include *default* Window Clause syntax!

<u>FIRSTNAME</u>	<u>GENDER</u>	<u>WEIGHT</u>	<u>LT_CHILD</u>	<u>HV_CHILD</u>
ROSEMARY	F	123	ROSEMARY	ROSEMARY
LAUREN	F	876	ROSEMARY	LAUREN
ALBERT	M	150	ALBERT	ALBERT
TOMMY	M	167	ALBERT	TOMMY
BUDDY	M	189	ALBERT	BUDDY
FARQUAR	M	198	ALBERT	FARQUAR
SIMON	M	256	ALBERT	SIMON

# Window Clause Syntax

*windowing\_clause*::=



- **ROWS** allows you to limit data by *rows*.
- **RANGE** allows you to limit data by *column value*, but is limited in SQL Server:
  - RANGE is only supported with UNBOUNDED and CURRENT ROW window frame delimiters.
- **The default Windowing Clause is ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW.**

## Example #15



- Task: Retrieve the name of the heaviest and lightest male and female child.
- Note: Include *appropriate* Window Clause syntax!

```
SELECT A.FIRSTNAME, A.GENDER, A.WEIGHT,  
       FIRST_VALUE(A.FIRSTNAME) OVER (PARTITION BY A.GENDER  
                                       ORDER BY A.WEIGHT  
                                       ROWS BETWEEN UNBOUNDED PRECEDING  
                                               AND UNBOUNDED FOLLOWING)  
                                       AS LT_CHILD,  
       LAST_VALUE(A.FIRSTNAME) OVER (PARTITION BY A.GENDER  
                                       ORDER BY A.WEIGHT  
                                       ROWS BETWEEN UNBOUNDED PRECEDING  
                                               AND UNBOUNDED FOLLOWING)  
                                       AS HV_CHILD  
FROM CHILDSTAT A  
ORDER BY A.GENDER, A.WEIGHT
```



## Example #15



- Task: Retrieve the name of the heaviest and lightest male and female child.
- Note: Include *appropriate* Window Clause syntax!

<u>FIRSTNAME</u>	<u>GENDER</u>	<u>WEIGHT</u>	<u>LT_CHILD</u>	<u>HV_CHILD</u>
ROSEMARY	F	123	ROSEMARY	LAUREN
LAUREN	F	876	ROSEMARY	LAUREN
ALBERT	M	150	ALBERT	SIMON
TOMMY	M	167	ALBERT	SIMON
BUDDY	M	189	ALBERT	SIMON
FARQUAR	M	198	ALBERT	SIMON
SIMON	M	256	ALBERT	SIMON

## Example #16



- Task: Motivational Example #2 – Revisited
- Running totals worked because of Window Clause!

```
SELECT A.GENDER,A.FIRSTNAME,A.WEIGHT,  
       SUM(A.WEIGHT) OVER (PARTITION BY A.GENDER ORDER BY A.WEIGHT  
                           ROWS BETWEEN UNBOUNDED PRECEDING  
                           AND CURRENT ROW) AS WT_RUN,  
       SUM(A.WEIGHT) OVER (PARTITION BY A.GENDER ORDER BY A.WEIGHT  
                           ROWS BETWEEN UNBOUNDED PRECEDING  
                           AND UNBOUNDED FOLLOWING) AS WT_RUN2  
FROM CHILDSTAT A  
ORDER BY A.GENDER,A.WEIGHT
```

<u>GENDER</u>	<u>FIRSTNAME</u>	<u>WEIGHT</u>	<u>WT_RUN</u>	<u>WT_RUN2</u>
F	ROSEMARY	123	123	999
F	LAUREN	876	999	999
M	ALBERT	150	150	960
M	TOMMY	167	317	960
M	BUDDY	189	506	960
M	FARQUAR	198	704	960
M	SIMON	256	960	960

## Example #17



- **Task:** Compute the average weight using current, previous and next rows.
- **Note:** Use the ROWS Window Clause.

```
SELECT A.FIRSTNAME,A.GENDER,A.WEIGHT,  
       AVG(A.WEIGHT) OVER (PARTITION BY A.GENDER  
                           ORDER BY A.WEIGHT  
                           ROWS BETWEEN 1 PRECEDING  
                                   AND 1 FOLLOWING) AS AVG_3  
FROM CHILDSTAT A  
ORDER BY A.GENDER,A.WEIGHT
```

<b>FIRSTNAME</b>	<b>GENDER</b>	<b>WEIGHT</b>	<b>AVG_3</b>
ROSEMARY	F	123	499.5
LAUREN	F	876	499.5
ALBERT	M	150	158.5
TOMMY	M	167	168.6
BUDDY	M	189	184.6
FARQUAR	M	198	214.3
SIMON	M	256	227

## Example #18



- **Task:** Compute the sum of weight based on a range of height between 10 less than the current row's value to 5 more than the current row's value.
- **Note:** Use the RANGE Window Clause.
- **Note:** Oracle-specific example!

```
SELECT A.FIRSTNAME,A.HEIGHT,A.WEIGHT,  
       SUM(A.WEIGHT) OVER (ORDER BY A.HEIGHT  
                           RANGE BETWEEN 10 PRECEDING  
                           AND 5 FOLLOWING) AS SUM_10_5  
FROM CHILDSTAT A  
ORDER BY A.HEIGHT
```

<u>FIRSTNAME</u>	<u>HEIGHT</u>	<u>WEIGHT</u>	<u>SUM_10_5</u>
ROSEMARY	35	123	123
BUDDY	45	189	462
ALBERT	45	150	462
LAUREN	54	876	1215
FARQUAR	76	198	365
TOMMY	78	167	365
SIMON	87	256	423

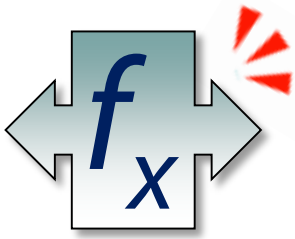
## Example #19



- **Task:** Compute average weight within a 90-day window of the child's birthdate.
- **Note:** Use the RANGE Window Clause.
- **Note:** Oracle-specific example!

```
SELECT A.FIRSTNAME, A.BIRTHDATE, A.WEIGHT,  
       AVG(A.WEIGHT) OVER (ORDER BY A.BIRTHDATE  
                           RANGE BETWEEN INTERVAL '30' DAY PRECEDING  
                           AND INTERVAL '60' DAY FOLLOWING)  
                           AS AVG_30_60  
FROM CHILDSTAT A  
ORDER BY A.BIRTHDATE
```

<u>FIRSTNAME</u>	<u>BIRTHDATE</u>	<u>WEIGHT</u>	<u>AVG_30_60</u>
BUDDY	02-OCT-98	189	193.5
FARQUAR	05-NOV-98	198	207
TOMMY	11-DEC-98	167	211.5
SIMON	03-JAN-99	256	211.5
ROSEMARY	08-MAY-00	123	499.5
LAUREN	10-JUN-00	876	513
ALBERT	02-AUG-00	150	150

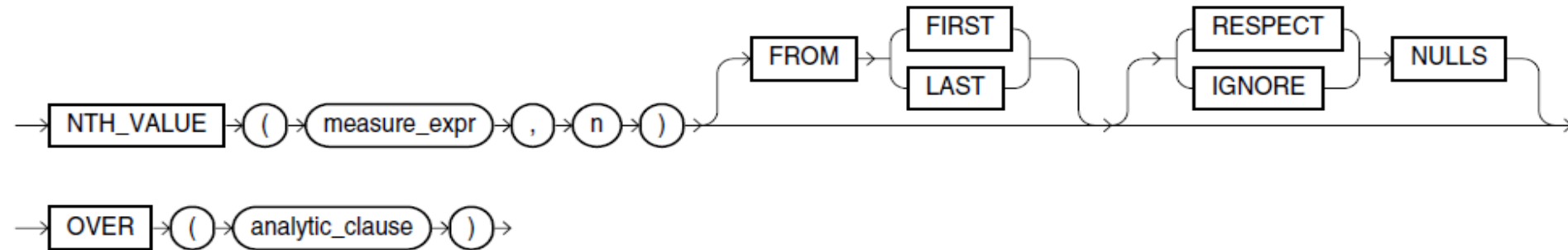


# NTH\_VALUE() Function

- **What is the NTH\_VALUE() Analytic Function?**
  - Generalization of FIRST\_VALUE() and LAST\_VALUE()
  - FIRST\_VALUE() returns the first row in the partition
  - LAST\_VALUE() returns the last row in the partition
  - NTH\_VALUE() returns the desired row in the partition.
  - ORDER BY Required
  - Can retrieve data from "top" or "bottom" of table
- **Availability:**
  - Oracle: 11g/R2
  - SQL Server: N/A

# NTH\_VALUE() Function

- What is the NTH\_VALUE() Analytic Function?
- Syntax:



- *measure\_expr* – usually a column name
- *n* – the row within the data you want returned
- FROM FIRST (default) pulls row starting from top on down
- FROM LAST pulls row starting from bottom on up
- RESPECT NULLS (default) includes NULLs, IGNORE NULLS excludes NULLs
- Probably want ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING

## Example #20



- Task: Determine the weight one heavier than the minimum weight and the weight one lighter than the maximum weight.
- Note: Specify correct Window Clause!

```
SELECT A.FIRSTNAME,A.GENDER,A.WEIGHT,  
       NTH_VALUE(A.WEIGHT,2) FROM FIRST  
         OVER (PARTITION BY A.GENDER  
              ORDER BY A.WEIGHT  
              ROWS BETWEEN UNBOUNDED PRECEDING  
                      AND UNBOUNDED FOLLOWING)  
                                     AS NEXT_HEAVY_GDR,  
       NTH_VALUE(A.WEIGHT,2) FROM LAST  
         OVER (PARTITION BY A.GENDER  
              ORDER BY A.WEIGHT  
              ROWS BETWEEN UNBOUNDED PRECEDING  
                      AND UNBOUNDED FOLLOWING)  
                                     AS NTOLAST_HEAVY_GDR  
FROM CHILDSTAT A
```



## Example #20



- **Task:** Determine the weight one heavier than the minimum weight and the weight one lighter than the maximum weight.
- **Note:** Specify correct Window Clause!

<u>FIRSTNAME</u>	<u>GENDER</u>	<u>WEIGHT</u>	<u>NEXT_HEAVY_GDR</u>	<u>NTOLAST_HEAVY_GDR</u>
ROSEMARY	F	123	876	123
LAUREN	F	876	876	123
ALBERT	M	150	167	198
TOMMY	M	167	167	198
BUDDY	M	189	167	198
FARQUAR	M	198	167	198
SIMON	M	256	167	198

# Summary

- **Window Clause allows fine-grained access**
- **Limit data by ROWS or RANGE**
- **Don't forget the default:**
  - ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW