Advanced SQL in Oracle and SQL Server

Analytic Functions – Part IV

Scott L. Hecht
http://www.sheepsqueezers.com
@sheepsqueezers

pluralsight
hardcore developer training

Module Contents

- **Analytic Functions**
  - Data used in Module
  - KEEP Clause
  - Statistics-Related Analytic Functions
    - MEDIAN()
    - NTILE()
    - PERCENT_RANK()
    - CUME_DIST()
    - PERCENTILE_DISC()
    - PERCENTILE_CONT()
  - 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**

FIRSTNAME	GENDER	BIRTHDATE	HEIGHT	WEIGHT
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

FIRST/LAST with the KEEP Clause

- **What is the KEEP Clause?**
  - Funnels subset of data based, not on partitions or windows, but on a function.
    - Only DENSE_RANK() is available, but maybe more in the future!
  - Not FIRST_VALUE() and LAST_VALUE()!
  - ORDER BY Required
  - OVER Clause is not required
  - OVER excluded → function behaves in an aggregate sense
  - OVER included → function behaves in an analytic sense
  - Availability:
    - Oracle: 9i/R1
    - SQL Server: N/A

FIRST/LAST with the KEEP Clause

FIRST/LAST with the KEEP Clause

- **Reminder of DENSE_RANK()**
  - Recall that RANK() returns a discontiguous series of values: 1, 1, 3, 4, …
  - DENSE_RANK(), though, returns a contiguous series of values: 1, 1, 2, 3, …
- **The first value returned by DENSE_RANK(), the "1", is associated with the FIRST keyword.**
- **The last value returned by DENSE_RANK() is associated with the LAST keyword.**
- **Only FIRST/LAST rows funneled into function.**

FIRST/LAST with the KEEP Clause

- **Syntax:**

```
function(…) KEEP (DENSE_RANK FIRST | LAST
                  ORDER BY var1,var2,…)
            OVER ( … )
```

  - *function* can, for the most part, be one of the standard aggregate functions: SUM(), MIN(), MAX(), AVG(), etc.
  - The keyword KEEP indicates that you intend to subset the data either by using the FIRST or the LAST keywords along with the required DENSE_RANK keyword.
  - Must use the ORDER BY Clause

Example #22

- Task: Use DENSE_RANK() on the height and partition by gender.
- Note: This data used in subsequent examples.

```
SELECT A.FIRSTNAME,A.HEIGHT,
       DENSE_RANK() OVER (PARTITION BY A.GENDER
                          ORDER BY A.HEIGHT) AS HEIGHT_DENSERANK
 FROM CHILDSTAT A
 ORDER BY A.GENDER,A.HEIGHT
```

FIRSTNAME	GENDER	HEIGHT	HEIGHT_DENSERANK
ROSEMARY	F	35	1
LAUREN	F	54	2
ALBERT	M	45	1
BUDDY	M	45	1
FARQUAR	M	76	2
TOMMY	M	78	3
SIMON	M	87	4

Example #23

- **Task: What is the average weight of the shortest males/females?**

```
SELECT A.FIRSTNAME,A.GENDER,A.WEIGHT,A.HEIGHT,
       DENSE_RANK() OVER (PARTITION BY A.GENDER
                          ORDER BY A.HEIGHT) AS HEIGHT_DENSERANK,
       AVG(A.WEIGHT) KEEP (DENSE_RANK FIRST ORDER BY A.HEIGHT)
                     OVER (PARTITION BY A.GENDER) AS AVG_WT
 FROM CHILDSTAT A
 ORDER BY A.GENDER,A.HEIGHT
```

FIRSTNAME	GENDER	WEIGHT	HEIGHT	HEIGHT_DENSERANK	AVG_WT
ROSEMARY	F	123	35	**1**	123
LAUREN	F	876	54	2	123
ALBERT	M	150	45	**1**	169.5
BUDDY	M	189	45	**1**	169.5
FARQUAR	M	198	76	2	169.5
TOMMY	M	167	78	3	169.5
SIMON	M	256	87	4	169.5

Example #24

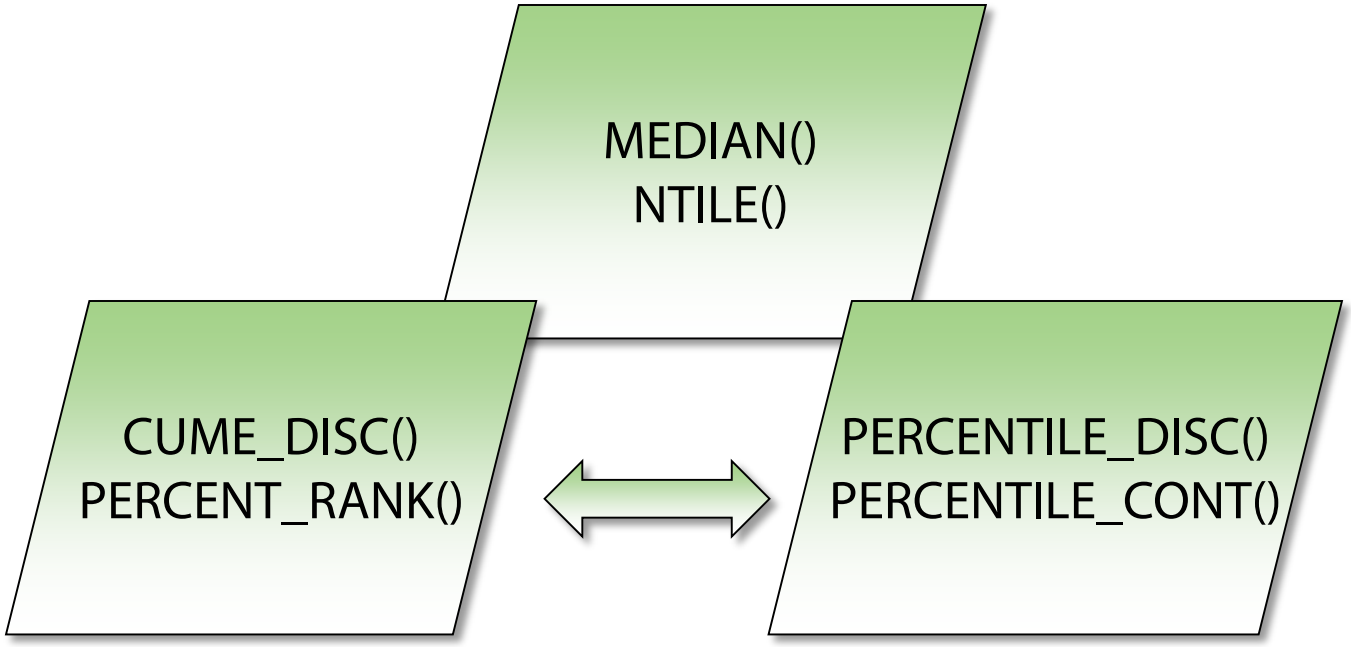
- **Task: What is the average weight of the tallest males/females?**

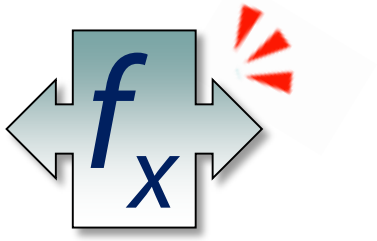
```
SELECT A.FIRSTNAME,A.GENDER,A.WEIGHT,A.HEIGHT,
       DENSE_RANK() OVER (PARTITION BY A.GENDER
                          ORDER BY A.HEIGHT) AS HEIGHT_DENSERANK,
       AVG(A.WEIGHT) KEEP (DENSE_RANK LAST ORDER BY A.HEIGHT)
                     OVER (PARTITION BY A.GENDER) AS AVG_WT
 FROM CHILDSTAT A
 ORDER BY A.GENDER,A.HEIGHT
```

FIRSTNAME	GENDER	WEIGHT	HEIGHT	HEIGHT_DENSERANK	AVG_WT
ROSEMARY	F	123	35	1	876
LAUREN	F	876	54	**2**	876
ALBERT	M	150	45	1	256
BUDDY	M	189	45	1	256
FARQUAR	M	198	76	2	256
TOMMY	M	167	78	3	256
SIMON	M	256	87	**4**	256

Statistics-Related Analytic Functions

- **Statistics-Related Analytic Functions**
  - MEDIAN() – computes the median (Oracle-specific)
  - NTILE() – splits rows into a specified number of buckets
  - CUME_DIST & PERCENT_RANK() – given a value, returns percentile
  - PERCENTILE_DISC() & PERCENTILE_CONT() – given a percentile, returns value
  - Some can be used in an aggregate sense as well!





MEDIAN() Function

- **What is the MEDIAN() function?**
  - The middle value of the ordered data
  - If odd number of rows, return the middle value
  - If even number of rows, average the two middle values
  - Can be used in an aggregate sense
  - Availability:
    - Oracle: 10g/R1
    - SQL Server: N/A
- **Syntax:**

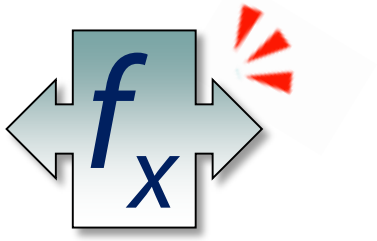
```
MEDIAN(column) OVER ( … )
```

Example #25

- **Task: Determine the median weight by gender.**

```
SELECT A.FIRSTNAME,A.GENDER,A.WEIGHT,
       MEDIAN(A.WEIGHT) OVER (PARTITION BY A.GENDER) AS MEDIAN_WT
 FROM CHILDSTAT A
 ORDER BY A.GENDER,A.WEIGHT
```

FIRSTNAME	GENDER	WEIGHT	MEDIAN_WT
ROSEMARY	F	123	499.5
LAUREN	F	876	499.5
ALBERT	M	150	189
TOMMY	M	167	189
BUDDY	M	189	189
FARQUAR	M	198	189
SIMON	M	256	189



NTILE() Function

- **What is the NTILE() function?**
  - Single parameter indicates number of desired buckets
  - Returns an integer representing group inclusion of each row
  - Groups are computed based (approx.) on the CEIL(*#rows*/*#groups*):
    - NTILE(4) for 7 row table
      - 7/4 =1.75 → 2 (each bucket contains 2 rows, except for last bucket)
      - Results: 1,1,2,2,3,3,4
  - Attempts to fill each bucket with the same number of rows
  - Assumes you have enough data
  - Availability:
    - Oracle: 8i
    - SQL Server: 2005
- **Syntax:**

```
NTILE(value) OVER ( … ORDER BY col1,col2, … )
```

Example #26

- **Task: Break up the height into quartiles.**

```
SELECT A.FIRSTNAME,A.HEIGHT,
       NTILE(4) OVER (ORDER BY A.HEIGHT) AS GRP4_HT
 FROM CHILDSTAT A
 ORDER BY A.HEIGHT
```

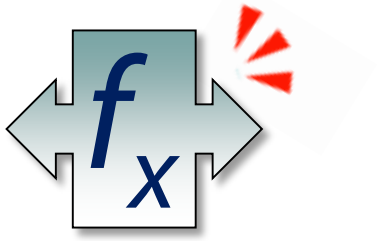
FIRSTNAME	HEIGHT	GRP4_HT
ROSEMARY	35	1
ALBERT	45	1
BUDDY	45	2
LAUREN	54	2
FARQUAR	76	3
TOMMY	78	3
SIMON	87	4

Example #27

- **Task: Break up the height into four groups by gender.**

```
SELECT A.FIRSTNAME,A.GENDER,A.HEIGHT,
       NTILE(4) OVER (PARTITION BY A.GENDER
                      ORDER BY A.HEIGHT) AS GRP4_HT
 FROM CHILDSTAT A
 ORDER BY A.GENDER,A.HEIGHT
```

FIRSTNAME	GENDER	HEIGHT	GRP4_HT
ROSEMARY	F	35	1
LAUREN	F	54	2
ALBERT	M	45	1
BUDDY	M	45	1
FARQUAR	M	76	2
TOMMY	M	78	3
SIMON	M	87	4



CUME_DIST() Function

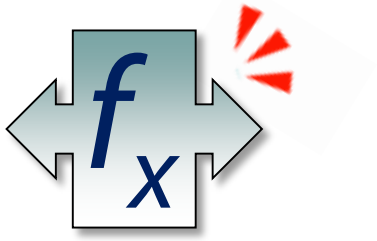
- **What is the CUME_DIST() function?**
  - CUME_DIST() is the **number of rows** with values less than or equal to that row's value divided by the **total number of rows**
  - **Approximate** formula: *row_number/total_rows*
  - Ranges from >0 to 1
  - Repeated column values receive the same CUME_DIST() value
  - Availability:
    - Oracle: 8i
    - SQL Server: 2012

Example #28

- **Task: Using CUME_DIST(), compute the cumulative distribution on the height.**

```
SELECT A.FIRSTNAME,A.HEIGHT,
       CUME_DIST() OVER (ORDER BY A.HEIGHT) AS CUMDIST_HEIGHT
 FROM CHILDSTAT A
 ORDER BY A.HEIGHT
```

FIRSTNAME	HEIGHT	CUMDIST_HEIGHT
ROSEMARY	35	0.1429
ALBERT	45	0.4286
BUDDY	45	0.4286
LAUREN	54	0.5714
FARQUAR	76	0.7143
TOMMY	78	0.8571
SIMON	87	1.0000



PERCENT_RANK() Function

- **What is the PERCENT_RANK() function?**
  - PERCENT_RANK() computes the rank, using RANK() on the column, subtracts 1 and then divides by the number of rows minus 1
  - Returns the cumulative distribution value from 0 to 1
  - Exact formula: *(rank -1)* / *(total_rows -1)*
  - Repeated column values receive the same PERCENT_RANK() value
  - Availability:
    - Oracle: 8i
    - SQL Server: 2012
- **Syntax:**

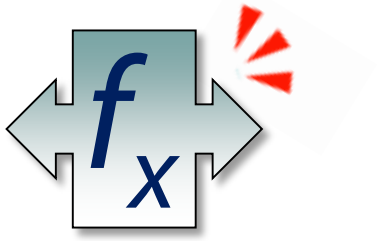
```
PERCENT_RANK() OVER ( … ORDER BY col1,col2,… )
```

Example #29

- **Task: Compute the percent rank on the height.**

```
SELECT A.FIRSTNAME,A.HEIGHT,
       RANK() OVER (ORDER BY A.HEIGHT) AS RANK_HEIGHT,
       PERCENT_RANK() OVER (ORDER BY A.HEIGHT) AS PCTDIST_HEIGHT
 FROM CHILDSTAT A
 ORDER BY A.HEIGHT
```

FIRSTNAME	HEIGHT	RANK_HEIGHT	PCTDIST_HEIGHT
ROSEMARY	35	1	0
ALBERT	45	2	0.1667
BUDDY	45	2	0.1667
LAUREN	54	4	0.5
FARQUAR	76	5	0.6667
TOMMY	78	6	0.8333
SIMON	87	7	1



PERCENTILE_DISC() Function

- **What is the PERCENTILE_DISC() function?**
  - Inverse of CUME_DIST()
  - Compares desired percentile to CUME_DIST() value. Returns column value associated with CUME_DIST() equal to or higher than desired percentile.
  - Values returned always from table
  - No interpolation performed
  - Availability:
    - Oracle: 9i/R1
    - SQL Server: 2012
- **Syntax:**

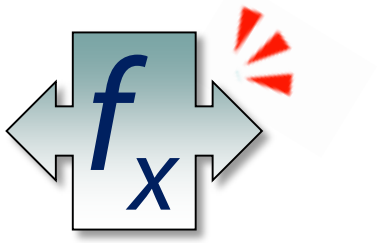
```
PERCENTILE_DISC(percentile)
  WITHIN GROUP (ORDER BY col1,col2,…) OVER ( … )
```

Example #30

- **Task: Compute the height associated with the percentiles .50 and .72.**

```
SELECT A.FIRSTNAME,A.HEIGHT,
       CUME_DIST() OVER (ORDER BY A.HEIGHT) AS CUMDIST_HEIGHT,
       PERCENTILE_DISC(.50) WITHIN GROUP (ORDER BY A.HEIGHT)
                                          OVER () AS PCTDISC_50_HT,
       PERCENTILE_DISC(.72) WITHIN GROUP (ORDER BY A.HEIGHT)
                                          OVER () AS PCTDISC_72_HT
 FROM CHILDSTAT A
 ORDER BY A.HEIGHT
```

FIRSTNAME	HEIGHT	CUMDIST_HEIGHT	PCTDISC_50_HT	PCTDISC_72_HT
ROSEMARY	35	0.1429	54	78
ALBERT	45	0.4286	54	78
BUDDY	45	0.4286	54	78
LAUREN	54	0.5714	54	78
FARQUAR	76	0.7143	54	78
TOMMY	78	0.8571	54	78
SIMON	87	1	54	78



PERCENTILE_CONT() Function

- **What is the PERCENTILE_CONT() function?**
  - Similar to PERCENTILE_DISC() except performs linear interpolation
  - Values returned are not necessarily from table
  - Determines row based on 1+*percentile**(*totalrows*-1)
    - First row determined by FLOOR(1+*percentile**(*totalrows*-1))
    - Second row determined by CEIL(1+*percentile**(*totalrows*-1))
  - Availability:
    - Oracle: 9i/R1
    - SQL Server: 2012
- **Syntax:**

```
PERCENTILE_CONT(percentile)
 WITHIN GROUP (ORDER BY col1,col2,…) OVER ( … )
```

Example #31

- **Task: Compute the height associated with the percentiles .50 and .72.**

```
SELECT A.FIRSTNAME,A.HEIGHT,
       PERCENTILE_CONT(.50) WITHIN GROUP (ORDER BY A.HEIGHT)
                                          OVER () AS PCTCONT_50_HT,
       PERCENTILE_CONT(.72) WITHIN GROUP (ORDER BY A.HEIGHT)
                                          OVER () AS PCTCONT_72_HT
 FROM CHILDSTAT A
 ORDER BY A.HEIGHT
```

FIRSTNAME	HEIGHT	PCTCONT_50_HT	PCTCONT_72_HT
ROSEMARY	35	54	76.64
ALBERT	45	54	76.64
BUDDY	45	54	76.64
LAUREN	54	54	76.64
FARQUAR	76	54	76.64
TOMMY	78	54	76.64
SIMON	87	54	76.64

Summary

- KEEP Clause gives you another way to access data
- FIRST/LAST Keywords returns specific data
- Can even perform statistics!