Advanced SQL in Oracle and SQL Server

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

pluralsight
hardcore developer training

Course Contents

- Analytic Functions
- Extensions to GROUP BY
- The WITH Clause
- The PIVOT and UNPIVOT Features
- The MERGE Statement
- The PARTITION BY/RIGHT OUTER JOIN Feature

Software Requirements

- **Oracle Database**
  - Versions: 8i, 9i, 10g, 11g, 12c
  - Download from Oracle's website (http://www.oracle.com)
  - Client: Oracle SQL Developer, Toad, SQL*Plus command prompt, etc.
- **Microsoft SQL Server Database**
  - Versions: 2005, 2008, 2010, 2012
  - SQL Server 2014 is in Community Technology Preview (CTP)
  - Download from Microsoft's website (http://www.microsoft.com)
  - Client: SQL Server Management Studio

Advanced SQL
in Oracle and SQL Server

Analytic Functions – Part I

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

pluralsight
hardcore developer training

Module Contents

- **Analytic Functions**
  - □ Data used in Module
  - □ What are Analytic Functions?
  - □ Motivational Examples
  - □ PARTITION BY Clause
  - □ Summary

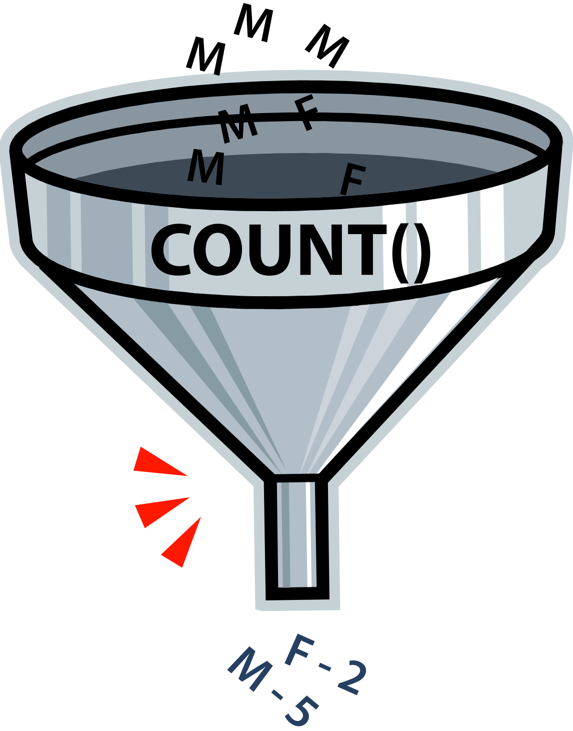
Data Used in Module

- **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**

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

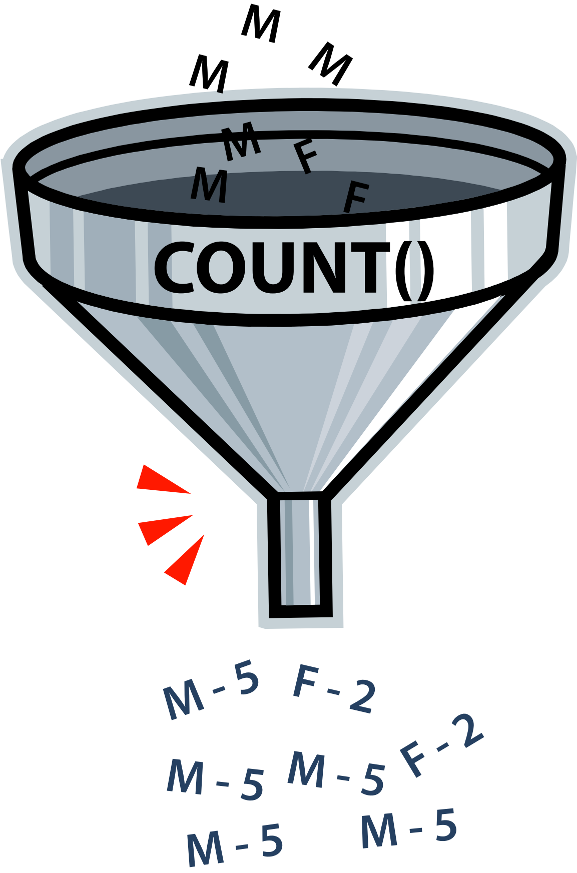
What are Analytic Functions?

- **Recall *Aggregate* Functions**
  - COUNT(), SUM(),… (used in an *aggregate sense*)
  - Performed, usually, along with a GROUP BY Clause
  - Resulting rows, usually, less than incoming



What are Analytic Functions?

- *Analytic* Functions
  - Perform GROUP BY-like summarizations
  - Access data in non-sequential way
  - COUNT(), SUM(), … (used in an *analytic sense*)
  - Resulting row count **same** as incoming
  - Simplify SQL code (no join)
  - Eliminates intermediate tables
  - Also known as Window Functions



What are Analytic Functions?

- *Analytic* **Functions**
  - **Last** SQL operation performed
  - Performed on results of SQL query
  - Not affected by:
    - GROUP BY
    - HAVING
    - WHERE
- **Syntax**

```
function(…) OVER (PARTITION BY col1,col2,…
                  ORDER BY col3,col4,…
                  …windowing-clause…) AS column-name
```

Motivational Example #1

- **Non-Analytic Method**
  - Task: Create a column containing row counts within gender.
  - Step 1: First, create table holding counts by gender.

```
CREATE TABLE CHILDSTAT_COUNT_BY_GENDER AS
 SELECT GENDER,COUNT(*) AS GENDER_COUNTS
  FROM CHILDSTAT
  GROUP BY GENDER
```

```
GENDER GENDER_COUNTS
F      2
M      5
```

Motivational Example #1

- **Non-Analytic Method (*continued*)**
  - Task: Create a column containing row counts within gender.
  - Step 1: Create table holding counts by gender.
  - Step 2: Merge against CHILDSTAT.

```
SELECT A.*,B.GENDER_COUNTS
 FROM CHILDSTAT A INNER JOIN CHILDSTAT_COUNT_BY_GENDER B
 ON A.GENDER=B.GENDER
```

```
FIRSTNAME  GENDER BIRTHDATE  HEIGHT  WEIGHT  GENDER_COUNTS
ROSEMARY   F      08-MAY-00  35      123     2
LAUREN     F      10-JUN-00  54      876     2
ALBERT     M      02-AUG-00  45      150     5
BUDDY      M      02-OCT-98  45      189     5
FARQUAR    M      05-NOV-98  76      198     5
TOMMY      M      11-DEC-98  78      167     5
SIMON      M      03-JAN-99  87      256     5
```

Motivational Example #1

- **Analytic Method**
  - Task: Create a column containing row counts within gender.

```
SELECT A.*,
       COUNT(*) OVER (PARTITION BY A.GENDER) AS GENDER_COUNTS
 FROM CHILDSTAT A
```

```
FIRSTNAME  GENDER BIRTHDATE  HEIGHT  WEIGHT  GENDER_COUNTS
ROSEMARY   F      08-MAY-00  35      123     2
LAUREN     F      10-JUN-00  54      876     2
-----------------------------------------------------------
ALBERT     M      02-AUG-00  45      150     5
BUDDY      M      02-OCT-98  45      189     5
FARQUAR    M      05-NOV-98  76      198     5
TOMMY      M      11-DEC-98  78      167     5
SIMON      M      03-JAN-99  87      256     5
```

Motivational Example #2

- **Analytic Method**
  - Task: Create running totals of weight by gender.

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

```
GENDER FIRSTNAME WEIGHT WT_RUN
------ --------- ------ ------
F      ROSEMARY  123    123
F      LAUREN    876    999
------------------------------
M      ALBERT    150    150
M      TOMMY     167    317
M      BUDDY     189    506
M      FARQUAR   198    704
M      SIMON     256    960
```

PARTITION BY Clause

- **What is the PARTITION BY Clause?**
    - Also called *Query Partition Clause*
    - Similar to the GROUP BY Clause
        - Breaks up data into chunks (or *partitions*)
        - Separated by *partition boundary*
        - Function performed within partitions
        - Re-initialized when crossing partition boundary
- **Syntax**

```
function(…) OVER (PARTITION BY col1,col2,…)
```

- **Functions**
    - Familiar functions such as COUNT(), SUM(), MIN(), MAX(), etc.
    - New functions as well (e.g., ROW_NUMBER(), RATIO_TO_REPORT(), etc.)

Example #1

- **Task: Create a column containing row counts within gender.**

```
SELECT A.*,
       COUNT(*) OVER (PARTITION BY A.GENDER) AS GENDER_COUNTS
 FROM CHILDSTAT A
```

```
FIRSTNAME  GENDER BIRTHDATE  HEIGHT  WEIGHT  GENDER_COUNTS
ROSEMARY   F      08-MAY-00  35      123     2
LAUREN     F      10-JUN-00  54      876     2
ALBERT     M      02-AUG-00  45      150     5
BUDDY      M      02-OCT-98  45      189     5
FARQUAR    M      05-NOV-98  76      198     5
TOMMY      M      11-DEC-98  78      167     5
SIMON      M      03-JAN-99  87      256     5
```

Example #2

- **Task: Create a column containing maximum height within gender.**

```
SELECT A.*,
       MAX(A.HEIGHT) OVER (PARTITION BY A.GENDER) AS MAX_HT
 FROM CHILDSTAT A
```

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

Example #3

- Task: Create a column containing distinct height counts within gender.
- Can use DISTINCT keyword (Oracle, Not SQL Server).

```
SELECT A.*,
       COUNT(DISTINCT A.HEIGHT) OVER (PARTITION BY A.GENDER)
                                                       AS DIST_HT
 FROM CHILDSTAT A
```

```
FIRSTNAME  GENDER BIRTHDATE  HEIGHT  WEIGHT  DIST_HT
ROSEMARY   F      08-MAY-00  35      123     2
LAUREN     F      10-JUN-00  54      876     2
----------------------------------------------------
ALBERT     M      02-AUG-00  45      150     4
BUDDY      M      02-OCT-98  45      189     4
FARQUAR    M      05-NOV-98  76      198     4
TOMMY      M      11-DEC-98  78      167     4
SIMON      M      03-JAN-99  87      256     4
```

Example #4

- **Task: Create a column containing distinct number of genders.**
- **Note: Can use DISTINCT keyword (Oracle, Not SQL Server).**

```
SELECT A.*,
       COUNT(DISTINCT A.GENDER) OVER () AS DIST_GENDER
 FROM CHILDSTAT A
```

```
FIRSTNAME  GENDER BIRTHDATE  HEIGHT  WEIGHT  DIST_GENDER
LAUREN     F      10-JUN-00  54      876     2
ROSEMARY   F      08-MAY-00  35      123     2
FARQUAR    M      05-NOV-98  76      198     2
TOMMY      M      11-DEC-98  78      167     2
SIMON      M      03-JAN-99  87      256     2
ALBERT     M      02-AUG-00  45      150     2
BUDDY      M      02-OCT-98  45      189     2
```

RATIO_TO_REPORT() Function

- **RATIO_TO_REPORT() Analytic Function**
  - Computes ratio of a value to within-group total
  - Not a percent!
  - Sum of ratios add to 1
  - Availability:
    - Oracle: 8i
    - SQL Server: N/A
- **Syntax**

```
RATIO_TO_REPORT(column) OVER ( … )
```

Example #5

- **Task: Create percent of weight off total weight within gender.**

```
 SELECT A.*,
        100*RATIO_TO_REPORT(WEIGHT) OVER (PARTITION BY A.GENDER)
                                                  AS PCT_WT_GENDER
  FROM CHILDSTAT A
```

```
FIRSTNAME  GENDER BIRTHDATE  HEIGHT  WEIGHT  PCT_WT_GENDER
LAUREN     F      10-JUN-00  54      876     87.69
ROSEMARY   F      08-MAY-00  35      123     12.31
FARQUAR    M      05-NOV-98  76      198     20.63
TOMMY      M      11-DEC-98  78      167     17.40
SIMON      M      03-JAN-99  87      256     26.67
ALBERT     M      02-AUG-00  45      150     15.63
BUDDY      M      02-OCT-98  45      189     19.69
```

Example #6

- **Task: Create a column containing row counts within gender and birth year.**
- **Note: Use EXTRACT() in Oracle, YEAR() in SQL Server.**

```
SELECT A.*,
       COUNT(*) OVER (PARTITION BY A.GENDER,YEAR(BIRTHDATE))
                                                 AS CNT_GBY
 FROM CHILDSTAT A
 ORDER BY A.GENDER,A.BIRTHDATE
```

```
FIRSTNAME  GENDER BIRTHDATE  HEIGHT  WEIGHT  CNT_GBY
ROSEMARY   F      08-MAY-00  35      123     2
LAUREN     F      10-JUN-00  54      876     2
BUDDY      M      02-OCT-98  45      189     3
FARQUAR    M      05-NOV-98  76      198     3
TOMMY      M      11-DEC-98  78      167     3
SIMON      M      03-JAN-99  87      256     1
ALBERT     M      02-AUG-00  45      150     1
```

Summary

- PARTITION BY Clause breaks rows into chunks
- Allows for within-chunk computations
- No reduction in data!