Advanced SQL in Oracle and SQL Server

Extensions to GROUP BY

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

pluralsight
hardcore developer training

Module Contents

- **Extensions to GROUP BY**
  - □ Introduction
  - □ Data Used in Module
  - □ A Comment about SQL Server 2005
  - □ Warning about Temporary Space
  - □ Motivational Example
  - □ GROUP BY GROUPING SETS
  - □ GROUP BY ROLLUP
  - □ GROUP BY CUBE
  - □ Composite Columns
  - □ Using Multiple Extensions
  - □ Useful Functions:
    - □ GROUPING()
    - □ GROUPING_ID()
    - □ GROUP_ID()
  - □ Summary

Introduction

- **Why Learn Extensions to GROUP BY?**
    - Typical GROUP BY summarizes down to existing column levels
        - Only existing data summarized
        - No *n-at-a-time* combinations of the columns are produced
    - Producing *n-at-a-time* combinations requires many SQL queries
        - Typically use UNIONs
        - Copy-and-paste can be a nightmare
        - Chance of error in one of the SQL queries is great
    - New Features
        - Simple SQL Syntax
        - Virtually eliminates coding errors
        - Provides for *n-at-a-time* combinations
        - Request only those combinations desired
        - Potential for Temporary Space Problems
    - Availability:
        - Oracle: 8i
        - SQL Server: 2008

Data Used in Module

- **Table**
  - CANDYBAR_CONSUMPTION_DATA
- **Columns**
  - CONSUMER_ID – unique identifier of a consumer
  - CANDYBAR_NAME – name of candy bar (e.g., MARS BAR, TWIX BAR, …)
  - SURVEY_YEAR – year of survey responses (e.g., 2009, 2010, …)
  - GENDER – gender of respondent (e.g., M=Male, F=Female)
  - OVERALL_RATING– rating of candy bar ranging from 1=Low to 10=High
  - NUMBER_BARS_CONSUMED – number of candy bars consumed during year
- **Data**

CONSUMER_ID	CANDYBAR_NAME	SURVEY_YEAR	GENDER	OVERALL_RATING	NUMBER_BARS_CONSUMED
1	MARS BAR	2009	M	10	252
1	MARS BAR	2010	M	10	352
1	MARS BAR	2011	M	10	452
1	TWIX BAR	2009	M	10	6
1	TWIX BAR	2010	M	7	60
1	TWIX BAR	2011	M	8	600

…continues on next slide…

Data Used in Module

- Data (*continued*)

CONSUMER_ID	CANDYBAR_NAME	SURVEY_YEAR	GENDER	OVERALL_RATING	NUMBER_BARS_CONSUMED
2	HERSHEY BAR	2009	F	5	2
2	HERSHEY BAR	2010	F	5	3
2	HERSHEY BAR	2011	F	5	1
2	MARS BAR	2009	F	8	25
2	MARS BAR	2010	F	8	12
2	MARS BAR	2011	F	8	13
3	MARS BAR	2009	M	8	25
3	MARS BAR	2010	M	7	12
3	MARS BAR	2011	M	8	13
3	TWIX BAR	2009	M	7	6
3	TWIX BAR	2010	M	8	60
3	TWIX BAR	2011	M	9	600
4	HERSHEY BAR	2009	F	7	20
4	HERSHEY BAR	2010	F	7	30
4	HERSHEY BAR	2011	F	7	10

…continues on next slide…

Data Used in Module

- Data (*continued*)

CONSUMER_ID	CANDYBAR_NAME	SURVEY_YEAR	GENDER	OVERALL_RATING	NUMBER_BARS_CONSUMED
4	MARS BAR	2009	F	7	25
4	MARS BAR	2010	F	7	35
4	MARS BAR	2011	F	7	15
4	TWIX BAR	2009	F	7	20
4	TWIX BAR	2010	F	7	30
4	TWIX BAR	2011	F	7	10
5	HERSHEY BAR	2009	M	8	15
5	HERSHEY BAR	2010	M	8	15
5	HERSHEY BAR	2011	M	6	5
5	SNICKERS BAR	2009	M	8	55
5	SNICKERS BAR	2010	M	8	65
5	SNICKERS BAR	2011	M	8	75
5	TWIX BAR	2009	M	9	75
5	TWIX BAR	2010	M	9	85
5	TWIX BAR	2011	M	9	95

A Comment about SQL Server 2005

- **Problem with DISTINCT Keyword**
  - Both CUBE and ROLLUP do not provide for the use of COUNT(DISTINCT *col1*), etc.
  - Error message you will receive:

> Distinct aggregates, for example AVG(DISTINCT *column_name*), COUNT(DISTINCT *column_name*), MAX(DISTINCT *column_name*), MIN(DISTINCT *column_name*), and SUM(DISTINCT *column_name*), are not supported when you use CUBE or ROLLUP. If they are used, the Microsoft SQL Server 2005 Database Engine returns an error message and cancels the query.

  - Problem resolved in SQL Server 2008+
  - No such problem exists in Oracle 8i+
  - SQL Server 2005 uses older WITH CUBE and WITH ROLLUP Syntax

A Warning about Temporary Space

- **Potential Issues with CUBE and ROLLUP**
  - All databases are a **shared** resource
  - SQL executing along with colleagues' SQL
  - Consuming CPU, Memory and Temporary Workspace
  - CUBE/ROLLUP produce many combinations of columns and can eat up a lot of temporary workspace!
  - When temporary workspace runs out, your SQL will bomb!
  - DBA can increase temporary workspace
  - Fix: Use only desired combinations
  - Fix: Use only desired rows of data
- **If SQL Bombs…**
  - E-Mail DBA with error message as well as your SQL code
  - Explain exactly what you are trying to do
  - Project/Priority
  - Sending a "My SQL bombed!" E-Mail will get you nowhere!
  - DO NOT JUST RE-SUBMIT YOUR SQL CODE IN THE HOPES IT WILL RUN THIS TIME!

Motivational Example

- **Combinations in the *Vintage* Style**
  - Let's use a GROUP BY to sum the NUMBER_BARS_CONSUMED to the SURVEY_YEAR, CANDYBAR_NAME,GENDER and OVERALL_RATING level.

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING
```

Motivational Example

- **Combinations in the *Vintage* Style (*continued*)**
  - Let's use a GROUP BY to sum the NUMBER_BARS_CONSUMED to the SURVEY_YEAR, CANDYBAR_NAME,GENDER and OVERALL_RATING level.

```
SURVEY_YEAR CANDYBAR_NAME GENDER OVERALL_RATING TOTAL_BARS_CONSUMED
2009        TWIX BAR      F      7              20
2009        MARS BAR      F      7              25
2011        HERSHEY BAR   F      5              1
2009        HERSHEY BAR   F      7              20
2009        SNICKERS BAR  M      8              55
2011        MARS BAR      M      10             452
2009        TWIX BAR      M      10             6
2009        MARS BAR      F      8              25
2011        SNICKERS BAR  M      8              75
2009        TWIX BAR      M      9              75
2010        HERSHEY BAR   M      8              15
2010        TWIX BAR      M      7              60
2009        HERSHEY BAR   F      5              2
…snip…
```

Motivational Example

- **Combinations in the *Vintage* Style (*continued*)**
  - Let's produce summary levels plus grand total

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING
UNION ALL
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,NULL AS OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY SURVEY_YEAR,CANDYBAR_NAME,GENDER
UNION ALL
SELECT SURVEY_YEAR,CANDYBAR_NAME,NULL AS GENDER,NULL AS OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY SURVEY_YEAR,CANDYBAR_NAME
UNION ALL
```

…continued on next slide…

Motivational Example

- **Combinations in the *Vintage* Style (*continued*)**
  - Let's produce summary levels plus grand total

```
…continued from previous slide…
SELECT SURVEY_YEAR,NULL AS CANDYBAR_NAME,NULL AS GENDER,NULL AS
       OVERALL_RATING, SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY SURVEY_YEAR
UNION ALL
SELECT NULL AS SURVEY_YEAR,NULL AS CANDYBAR_NAME,NULL AS GENDER,NULL AS
       OVERALL_RATING,SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
```

Motivational Example

- **Combinations in the *Vintage* Style (*continued*)**
  - Let's produce summary levels plus grand total

```
SURVEY_YEAR CANDYBAR_NAME GENDER OVERALL_RATING TOTAL_BARS_CONSUMED
2009        HERSHEY BAR   F      5              2
2009        HERSHEY BAR   F      7              20
2009        HERSHEY BAR   F                     22
2010        TWIX BAR                            235
2010                                            759
2011        HERSHEY BAR   F      5              1
2011        TWIX BAR      M                     1295
2011        TWIX BAR                            1305
2011                                            1889
                                                3174
…snip…
```

Motivational Example

- **Combinations in the *Modern* Style**
  - Let's produce summary levels plus grand total using GROUP BY ROLLUP
  - Output is the same!

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY ROLLUP(SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING)
```

GROUPING SETS

- **What are Grouping Sets?**
  - Select *exactly* which combinations you want
  - Does not, by default, produce a Grand Total
  - Used on the GROUP BY Clause
  - Uses the GROUPING SETS() Syntax
- **Syntax**

```
GROUP BY GROUPING SETS(A,B,C,…)
```

…is equivalent to…

```
GROUP BY A
UNION ALL
GROUP BY B
UNION ALL
GROUP BY C
    …
```

Example #1

- **Task: Use grouping sets to produce *almost* the results in the Motivational Example.**

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY GROUPING SETS(
                        (SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING),
                        (SURVEY_YEAR,CANDYBAR_NAME,GENDER),
                        (SURVEY_YEAR,CANDYBAR_NAME),
                        (SURVEY_YEAR)
                       )
```

- **Note that within each pair of parentheses is a single column, or a comma-delimited list of columns.**
- **Grand Total is missing. Use empty parentheses to add the grand total.**

Example #1

- **Task: Use grouping sets to produce *exactly* the results in the Motivational Example.**

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY GROUPING SETS(
                        (SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING),
                        (SURVEY_YEAR,CANDYBAR_NAME,GENDER),
                        (SURVEY_YEAR,CANDYBAR_NAME),
                        (SURVEY_YEAR),
                        ()
                       )
```

ROLLUP

- **What is a Rollup?**
  - □ Produces combinations useful for a rollup report
  - □ Does, by default, produce a Grand Total
  - □ Used on the GROUP BY Clause
  - □ Uses the ROLLUP() Syntax
- **Syntax**

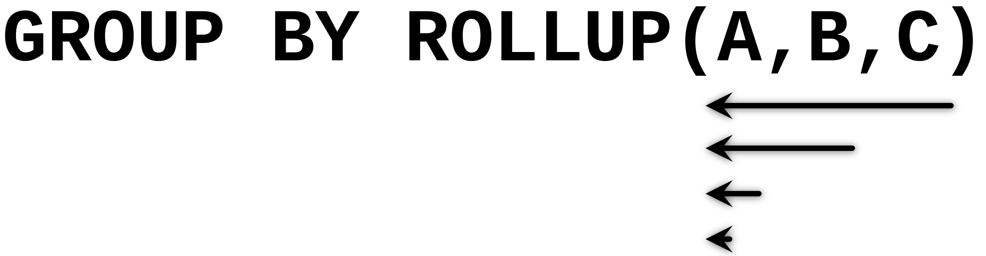
```
GROUP BY ROLLUP(A,B,C)
```

…is equivalent to…

```
GROUPING SETS(
               (A,B,C),
               (A,B),
               (A),
               ()
              )
```

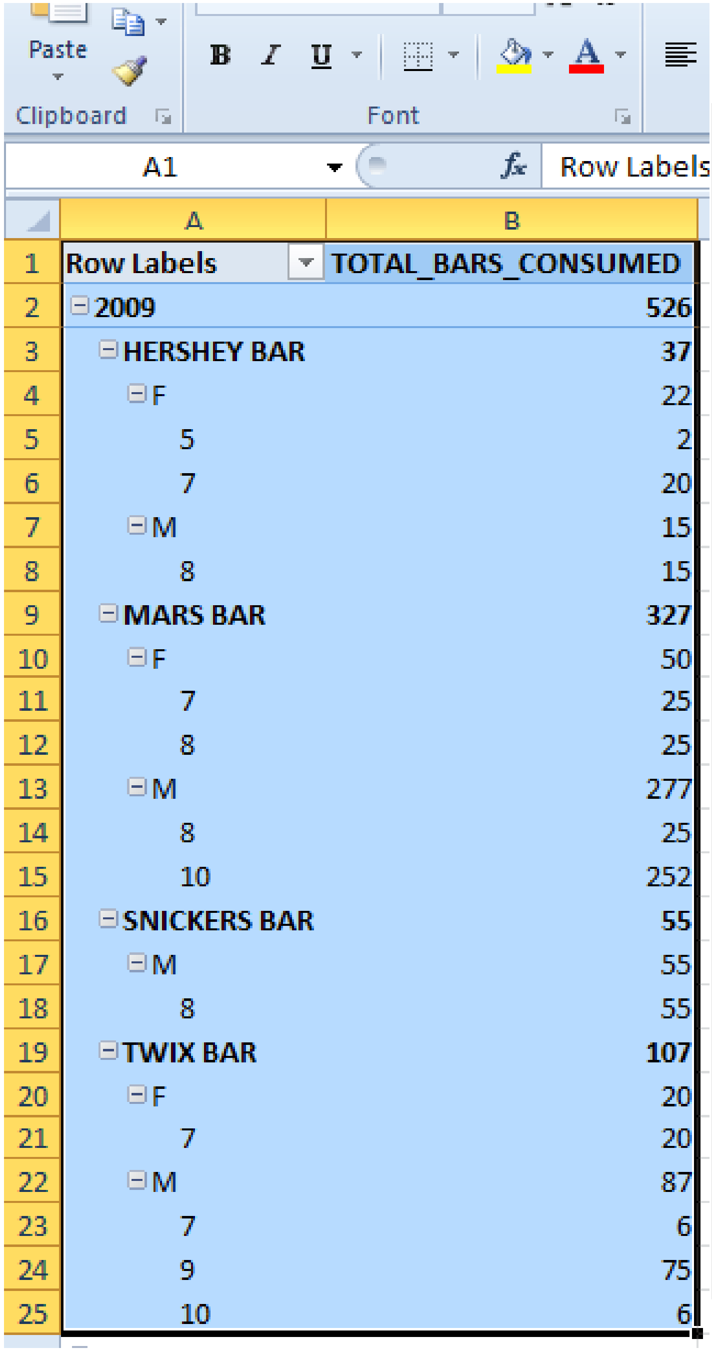
ROLLUP

- **What is a Rollup? (*continued*)**

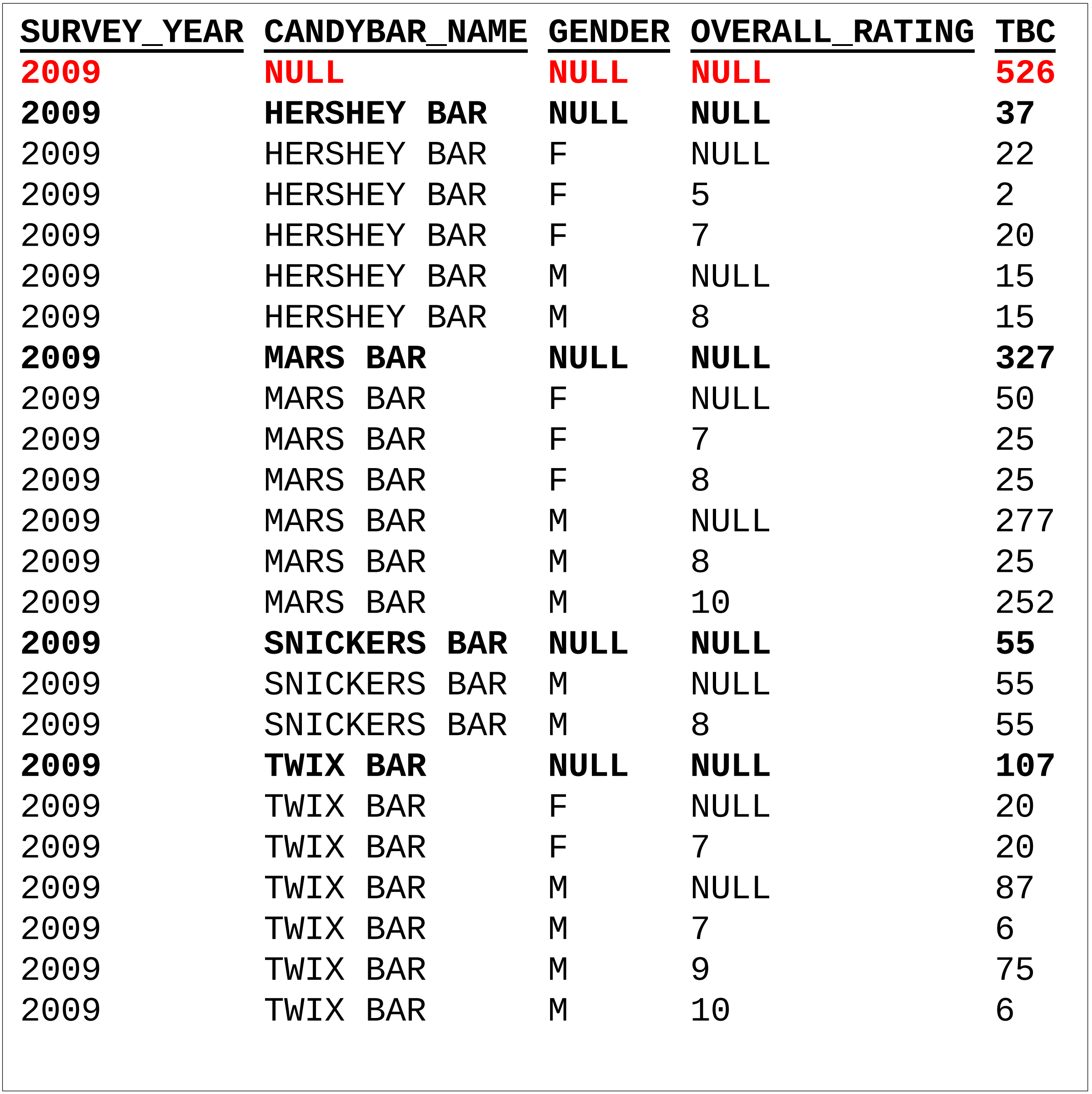


```
GROUP BY ROLLUP(A,B,C)
```

- **Used to Produce Rollup Reports**
  - □ See next slide



Row Labels	TOTAL_BARS_CONSUMED
⊟ 2009	**526**
⊟ HERSHEY BAR	**37**
⊟ F	22
5	2
7	20
⊟ M	15
8	15
⊟ MARS BAR	**327**
⊟ F	50
7	25
8	25
⊟ M	277
8	25
10	252
⊟ SNICKERS BAR	**55**
⊟ M	55
8	55
⊟ TWIX BAR	**107**
⊟ F	20
7	20
⊟ M	87
7	6
9	75
10	6

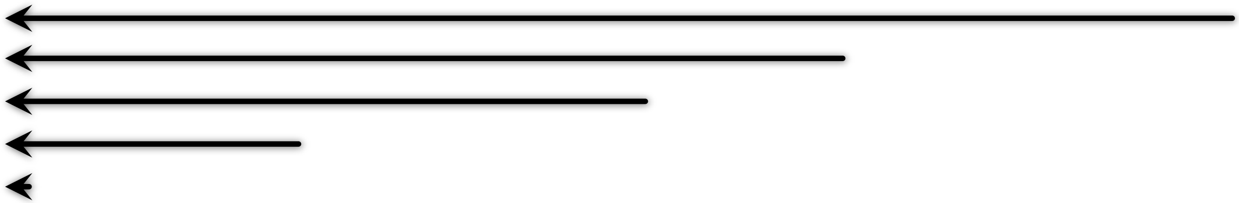


SURVEY_YEAR	CANDYBAR_NAME	GENDER	OVERALL_RATING	TBC
2009	**NULL**	**NULL**	**NULL**	**526**
2009	**HERSHEY BAR**	**NULL**	**NULL**	**37**
2009	HERSHEY BAR	F	NULL	22
2009	HERSHEY BAR	F	5	2
2009	HERSHEY BAR	F	7	20
2009	HERSHEY BAR	M	NULL	15
2009	HERSHEY BAR	M	8	15
2009	**MARS BAR**	**NULL**	**NULL**	**327**
2009	MARS BAR	F	NULL	50
2009	MARS BAR	F	7	25
2009	MARS BAR	F	8	25
2009	MARS BAR	M	NULL	277
2009	MARS BAR	M	8	25
2009	MARS BAR	M	10	252
2009	**SNICKERS BAR**	**NULL**	**NULL**	**55**
2009	SNICKERS BAR	M	NULL	55
2009	SNICKERS BAR	M	8	55
2009	**TWIX BAR**	**NULL**	**NULL**	**107**
2009	TWIX BAR	F	NULL	20
2009	TWIX BAR	F	7	20
2009	TWIX BAR	M	NULL	87
2009	TWIX BAR	M	7	6
2009	TWIX BAR	M	9	75
2009	TWIX BAR	M	10	6

Example #2

- **Task: Use ROLLUP to produce the exactly same results as the Motivational Example.**

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY ROLLUP(SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING)
```



CUBE

- **What is a Cube?**
  - Produces all combinations of columns 1-at-a-time, 2-at-a-time, etc.
  - Does, by default, produce a Grand Total
  - Used on the GROUP BY Clause
  - Uses the CUBE() Syntax
- **Syntax**

```
GROUP BY CUBE(A,B,C)
```

…is equivalent to…

```
GROUPING SETS(
               (A),(B),(C),
               (A,B),(A,C),(B,C),
               (A,B,C),
               ()
             )
```

Example #3

- **Task: Create all combinations of data using the variables SURVEY_YEAR, CANDYBAR_NAME, GENDER and OVERALL_RATING.**

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY CUBE(SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING)
```

```
SURVEY_YEAR CANDYBAR_NAME GENDER OVERALL_RATING TOTAL_BARS_CONSUMED
                                                3174
                                 5              6
                          F                     251
                          F      5              6
                          M                     2923
                          M      6              5
            SNICKERS BAR                        195
            SNICKERS BAR         8              195
            SNICKERS BAR  M                     195
            SNICKERS BAR  M      8              195
2009                                            526
2009                             5              2
2009                             7              71
2009                             8              120
2009                             9              75
2009                             10             258
2009                      F                     92
2009                      F      5              2
2009                      M                     434
2009                      M      7              6
2009        HERSHEY BAR                         37
2009        HERSHEY BAR          5              2
2009        HERSHEY BAR   F                     22
2009        HERSHEY BAR   F      5              2
```

…snip…

Composite Columns

- **What are Composite Columns?**
  - Two or more columns acting as one
  - Composite Columns skips combinations
  - Uses the familiar parenthesis syntax
- **Syntax Example**

```
GROUP BY ROLLUP(A,(B,C),D)
```

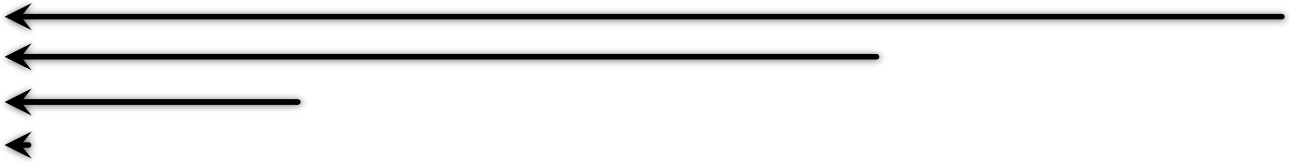
…is equivalent to…

```
GROUPING SETS(
               (A,B,C,D)
               (A,B,C),
               (A),
               ()
             )
```

Example #4

- **Task: Re-do the ROLLUP example, but ensure that CANDYBAR_NAME and GENDER act as one.**

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY ROLLUP(SURVEY_YEAR,(CANDYBAR_NAME,GENDER),OVERALL_RATING)
```



Using Multiple Extensions

- **What are Multiple Extensions?**
  - On a single GROUP BY, can specify:
    - One or more single columns
    - One or more ROLLUP
    - One or more CUBE
    - One or more GROUPING SETS
  - Allows for more varied summarizations
  - Caution: Can lead to repeated rows!

Example #5

- **Task: Re-do the CUBE example, but move SURVEY_YEAR from within the CUBE syntax.**

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY SURVEY_YEAR,
          CUBE(CANDYBAR_NAME,GENDER,OVERALL_RATING)
```

Example #5

- **Task: Re-do the CUBE example, but move SURVEY_YEAR from within the CUBE syntax.**

```
SURVEY_YEAR CANDYBAR_NAME GENDER OVERALL_RATING TOTAL_BARS_CONSUMED
2009                                            526
2009                             5              2
2009                             7              71
2009                             8              120
2009                             9              75
2009                             10             258
2009                      F                     92
2009                      F      5              2
2009                      F      7              65
2009                      F      8              25
2009                      M                     434
2009                      M      7              6
2009                      M      8              95
2009                      M      9              75
2009                      M      10             258
```

…continued on next slide…

Example #5

- **Task: Re-do the CUBE example, but move SURVEY_YEAR from within the CUBE syntax.**

```
SURVEY_YEAR CANDYBAR_NAME GENDER OVERALL_RATING TOTAL_BARS_CONSUMED
2009        MARS BAR                            327
2009        MARS BAR             7              25
2009        MARS BAR             8              50
2009        MARS BAR             10             252
2009        MARS BAR      F                     50
2009        MARS BAR      F      7              25
2009        MARS BAR      F      8              25
2009        MARS BAR      M                     277
2009        MARS BAR      M      8              25
2009        MARS BAR      M      10             252
2009        TWIX BAR                            107
2009        TWIX BAR             7              26
2009        TWIX BAR             9              75
2009        TWIX BAR             10             6
2009        TWIX BAR      F                     20
2009        TWIX BAR      F      7              20
2009        TWIX BAR      M                     87
2009        TWIX BAR      M      7              6
```

…continued on next slide…

Example #5

- **Task: Re-do the CUBE example, but move SURVEY_YEAR from within the CUBE syntax.**

```
SURVEY_YEAR CANDYBAR_NAME GENDER OVERALL_RATING TOTAL_BARS_CONSUMED
2009        TWIX BAR      M      9              75
2009        TWIX BAR      M      10             6
2009        HERSHEY BAR                         37
2009        HERSHEY BAR          5              2
2009        HERSHEY BAR          7              20
2009        HERSHEY BAR          8              15
2009        HERSHEY BAR   F                     22
2009        HERSHEY BAR   F      5              2
2009        HERSHEY BAR   F      7              20
2009        HERSHEY BAR   M                     15
2009        HERSHEY BAR   M      8              15
2009        SNICKERS BAR                        55
2009        SNICKERS BAR         8              55
2009        SNICKERS BAR  M                     55
2009        SNICKERS BAR  M      8              55
```

…snip…

Useful Functions – GROUPING()

- **What is the GROUPING() Function?**
  - Recall: NULLs represent the columns being summarized
  - Okay if your data has no NULLs
  - GROUPING(*column*) returns:
    - 1 – indicates the *column* is **not** being used in the GROUP BY (i.e., summary)
    - 0 – indicate the *column* is being used in the GROUP BY (i.e., actual)
- **Syntax**

```
GROUPING(column)
```

- **Availability:**
  - Oracle: 8i
  - SQL Server: 2005

Example #6

- Task: Re-do the ROLLUP example.
- Note: Use GROUPING() on each column in GROUP BY ROLLUP.

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TBC,
       GROUPING(SURVEY_YEAR) AS G_SY,
       GROUPING(CANDYBAR_NAME) AS G_CN,
       GROUPING(GENDER) AS G_G,
       GROUPING(OVERALL_RATING) AS G_OR
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY ROLLUP(SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING)
```

Example #6

- **Task: Re-do the ROLLUP example.**
- **Note: Use GROUPING() on each column in GROUP BY ROLLUP.**

```
SURVEY_YEAR CANDYBAR_NAME GENDER OVERALL_RATING TBC  G_SY G_CN G_G G_OR
                                                3174 1    1    1   1
2011                                            1889 0    1    1   1
2009        MARS BAR                            327  0    0    1   1
2009        MARS BAR      F                     50   0    0    0   1
2009        MARS BAR      M      8              25   0    0    0   0
…snip…
```

Useful Functions – GROUPING_ID()

- **What is the GROUPING_ID() Function?**
  - Recall: GROUPING() function used on single column
  - GROUPING_ID():
    - concatenates all the GROUPING() functions
    - binary to decimal conversion (e.g., $1111_2 \rightarrow 15_{10}$)
- **Syntax**

```
GROUPING_ID(col1,col2,col3,…)
```

- **Availability:**
  - Oracle: 9i/R1
  - SQL Server: 2008

Example #7

- Task: Re-do the ROLLUP example.
- Note: Use GROUPING_ID() on each column in GROUP BY ROLLUP.

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED,
       GROUPING_ID(SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING)
                                                                   AS GID
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY ROLLUP(SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING)
 ORDER BY 6 DESC
```

Example #7

- **Task: Re-do the ROLLUP example.**
- **Note: Use GROUPING_ID() on each column in GROUP BY ROLLUP.**

```
SURVEY_YEAR CANDYBAR_NAME GENDER OVERALL_RATING TBC  GID
                                                3174 15  = 1111
2011                                            1889 7   = 0111
2009        MARS BAR                            327  3   = 0011
2009        MARS BAR      F                     50   1   = 0001
2009        MARS BAR      M      8              25   0   = 0000
…snip…
```

Useful Functions – GROUP_ID()

- **What is the GROUP_ID() Function?**
  - Indicates which rows are duplicated
  - Useful when using multiple extensions together
  - GROUP_ID() returns
    - 0 – original row (i.e., not a repeated row)
    - 1, 2, … – indicates repeated rows
- **Syntax**

```
GROUP_ID()
```

- **Availability:**
  - Oracle: 9i/R1
  - SQL Server: N/A

Example #8

- Task: Re-do the ROLLUP example.
- Note: Use GROUP_ID() to determine repeated rows.

```
SELECT SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING,
       SUM(NUMBER_BARS_CONSUMED) AS TOTAL_BARS_CONSUMED,
       GROUP_ID() AS ROW_ID
 FROM CANDYBAR_CONSUMPTION_DATA
 GROUP BY SURVEY_YEAR,
          CUBE(SURVEY_YEAR,CANDYBAR_NAME,GENDER,OVERALL_RATING)
```

Example #8

- **Task: Re-do the ROLLUP example.**
- **Note: Use GROUP_ID() on each column in GROUP BY ROLLUP.**

```
SURVEY_YEAR CANDYBAR_NAME GENDER OVERALL_RATING TBC_  ROW_ID
2009                                            526   0
2010                                            759   0
2011                                            1889  0
2009                                            526   1
2010                                            759   1
2011                                            1889  1
…snip…
```

Summary

- Perform many GROUP BYs with ease
- GROUPING SETS() for those combinations you want
- ROLLUP() used for roll-up report
- CUBE() creates all combinations
- Multiple Extensions
- GROUPING(), GROUPING_ID(), GROUP_ID()