MTN.BI.03

Author: Volha Kutsevol
Lead Software Engineer
Volha_Kutsevol@epam.com

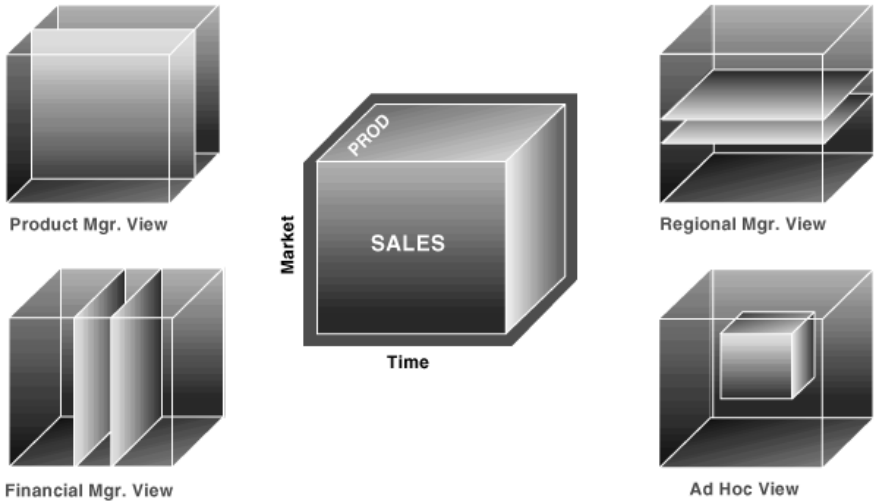
Oracle SQL for Aggregation in Data Warehouses

Agenda

1. Overview of SQL for Aggregation in Data Warehouses
2. SQL Grouping Extensions
   - 2.1 SQL GROUP BY Statement
   - 2.2 ROLLUP Extension to GROUP BY
   - 2.3 CUBE Extension to GROUP BY
   - 2.4 GROUPING Functions: GROUPING, GROUPING_ID, GROUP_ID
3. Grouping Sets
   - 3.1 Grouping Sets Expression
   - 3.2 Composite Columns
   - 3.3 Concatenated Groups

Overview of SQL for Aggregation in Data Warehouses

One of the most important advancements/features in the BI space is Oracle OLAP as part of the DB. But for users who do not have the luxury of using Oracle OLAP 11g in their environment, don't lose hope yet. ☺



Aggregation is a fundamental part of data warehousing. To improve aggregation performance in your warehouse, Oracle Database provides the following functionality:

- **CUBE** and **ROLLUP** extensions to the GROUP BY clause
- Three **GROUPING** functions
- **GROUPING SETS** expression

CUBE, ROLLUP, and GROUPING SETS extensions

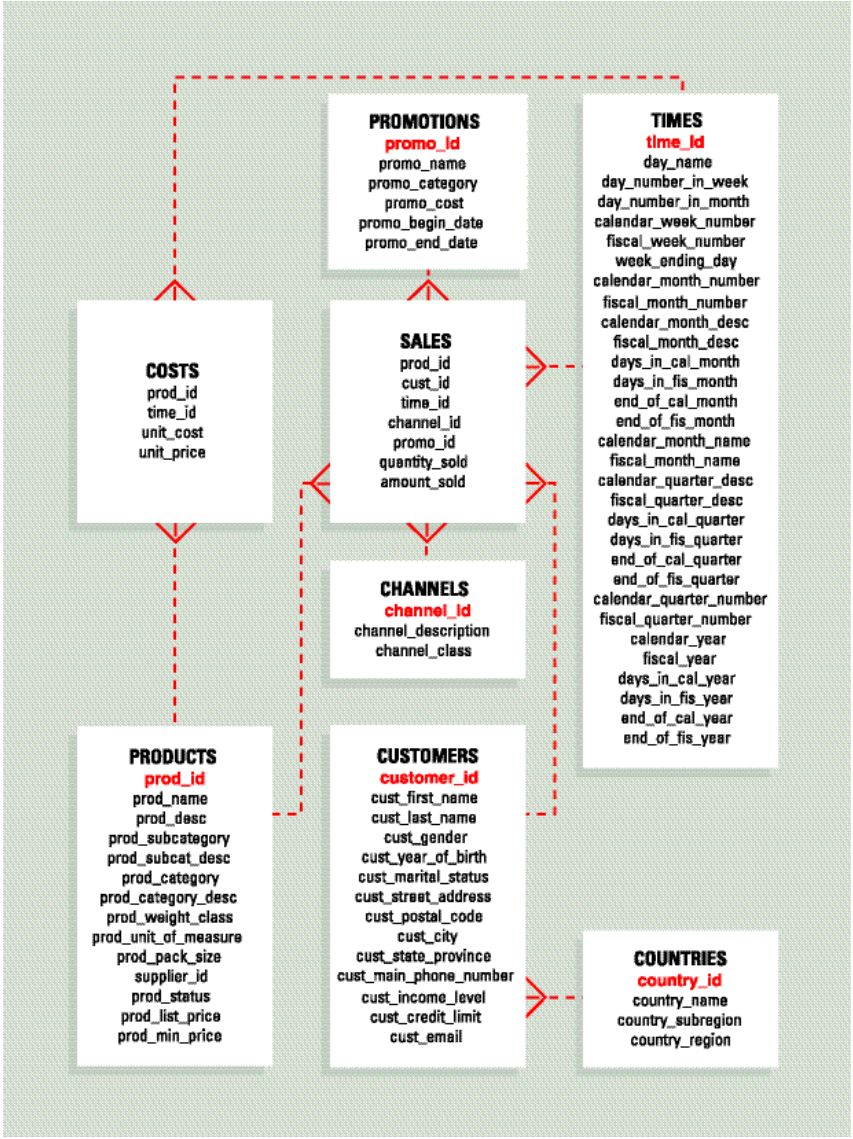
- The **CUBE, ROLLUP**, and **GROUPING SETS** extensions to SQL make querying and reporting easier and faster.

- **CUBE, ROLLUP,** and **GROUPING SETS** produce a single result set that is equivalent to a UNION ALL of differently grouped rows.

- The **CUBE, ROLLUP,** and the **GROUPING SETS** extension lets you specify just the groupings needed in the GROUP BY clause. This allows efficient analysis across multiple dimensions without performing a CUBE operation.

- To enhance performance, **CUBE, ROLLUP**, and **GROUPING SETS** can be parallelized: multiple processes can simultaneously execute all of these statements.

Sales History (SH)



SQL GROUP BY Statement

The **GROUP BY** statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

```
SELECT <column_name_1>,
       <column_name_2>,
aggregate_function(<column_name_N>)
FROM <table_name>
[WHERE condition]
GROUP BY <column_name_1>,
         <column_name_2>
```

SQL GROUP BY Statement

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     CUSTOMERS,
     TIMES,
     CHANNELS,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by TIMES.CALENDAR_MONTH_DESC,
         CHANNELS.CHANNEL_DESC,
         COUNTRIES.COUNTRY NAME
```

YEAR_MONTH	CHANNEL	COUNTRY	SALES$
2000-09	Direct Sales	United States of America	638,201
2000-09	Direct Sales	France	61,202
2000-09	Internet	France	9,597
2000-09	Internet	United States of America	124,224

SQL GROUP BY Statement

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     CUSTOMERS,
     TIMES,
     CHANNELS,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by TIMES.CALENDAR_MONTH_DESC,
         CHANNELS.CHANNEL_DESC,
         COUNTRIES.COUNTRY_NAME
order by 5
```

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$
2000-09	Internet	France	148	9,597
2000-09	Direct Sales	France	116	61,202
2000-09	Internet	United States of America	117	124,224
2000-09	Direct Sales	United States of America	95	638,201

ROLLUP Extension to GROUP BY

ROLLUP enables a SELECT statement to calculate multiple levels of subtotals across a specified group of dimensions. It also calculates a grand total. **ROLLUP** is a simple extension to the GROUP BY clause, so its syntax is extremely easy to use. The **ROLLUP** extension is highly efficient, adding minimal overhead to a query.

```
SELECT <column_name_1>,
       <column_name_2>,
aggregate_function(<column_name_N>)
FROM <table_name>
[WHERE condition]
GROUP BY ROLLUP(<column_name_1>,
                <column_name_2>)
```

ROLLUP Extension to GROUP BY

```
select COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     TIMES,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      TIMES.CALENDAR_MONTH_DESC='2000-09' AND
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by ROLLUP(COUNTRIES.COUNTRY NAME)
```

COUNTRY	AVG_SALES$	SALES$
France	105	2,112,221
United States of America	105	2,112,221
	105	**4,224,441**

ROLLUP extension produces group subtotals from right to left and a grand total. If "n" is the number of columns listed in the ROLLUP, there will be n+1 levels of subtotals:

ROLLUP (a, b, c):
(a, b, c)
(a, b)
(a)
()

ROLLUP Extension to GROUP BY

```
select COUNTRIES.COUNTRY_NAME COUNTRY,
        CHANNELS.CHANNEL_DESC CHANNEL,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     TIMES,
     COUNTRIES,
     CHANNELS
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      TIMES.CALENDAR_MONTH_DESC='2000-09' AND
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by ROLLUP(COUNTRIES.COUNTRY_NAME, CHANNELS.CHANNEL_DESC)
```

COUNTRY	CHANNEL	AVG_SALES$	SALES$
France	Internet	121	228,241
France	Partners	108	666,172
France	Direct Sales	102	1,217,808
France		**105**	**2,112,221**
United States of America	Internet	121	228,241
United States of America	Partners	108	666,172
United States of America	Direct Sales	102	1,217,808
United States of America		**105**	**2,112,221**
		105	**4,224,441**

ROLLUP Extension to GROUP BY

```
select COUNTRIES.COUNTRY_NAME COUNTRY,
       CHANNELS.CHANNEL_DESC CHANNEL,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     TIMES,
     COUNTRIES,
     CHANNELS
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      TIMES.CALENDAR_MONTH_DESC='2000-09' AND
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by ROLLUP(CHANNELS.CHANNEL_DESC, COUNTRIES.COUNTRY_NAME)
```

COUNTRY	CHANNEL	AVG_SALES$	SALES$
France	Internet	121	228,241
United States of America	Internet	121	228,241
	Internet	**121**	**456,482**
France	Partners	108	666,172
United States of America	Partners	108	666,172
	Partners	**108**	**1,332,343**
France	Direct Sales	102	1,217,808
United States of America	Direct Sales	102	1,217,808
	Direct Sales	**102**	**2,435,616**
		105	**4,224,441**

ROLLUP Extension to GROUP BY

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     CUSTOMERS,
     TIMES,
     CHANNELS,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by ROLLUP(TIMES.CALENDAR_MONTH_DESC,
                CHANNELS.CHANNEL_DESC,
                COUNTRIES.COUNTRY_NAME)
```

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$
2000-09	Internet	France	148	9,597
2000-09	Internet	United States of America	117	124,224
2000-09	**Internet**		**119**	**133,821**
2000-09	Direct Sales	France	116	61,202
2000-09	Direct Sales	United States of America	95	638,201
2000-09	**Direct Sales**		**96**	**699,403**
2000-09			**99**	**833,224**
			99	**833,224**

ROLLUP Extension to GROUP BY

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     CUSTOMERS,
     TIMES,
     CHANNELS,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by ROLLUP(TIMES.CALENDAR_MONTH_DESC,
                CHANNELS.CHANNEL_DESC,
                COUNTRIES.COUNTRY_NAME)
```

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$
2000-09	Internet	France	148	9,597
2000-09	Internet	United States of America	117	124,224
2000-09	**Internet**		**119**	**133,821**
2000-09	Direct Sales	France	116	61,202
2000-09	Direct Sales	United States of America	95	638,201
2000-09	**Direct Sales**		**96**	**699,403**
2000-09			**99**	**833,224**
			99	**833,224**

Partial ROLLUP

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     CUSTOMERS,
     TIMES,
     CHANNELS,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by TIMES.CALENDAR_MONTH_DESC,
         ROLLUP(CHANNELS.CHANNEL_DESC, COUNTRIES.COUNTRY_NAME )
```

Partial ROLLUP does not produce a grand total row.

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$
2000-09	Internet	France	148	9,597
2000-09	Internet	United States of America	117	124,224
2000-09	**Internet**		**119**	**133,821**
2000-09	Direct Sales	France	116	61,202
2000-09	Direct Sales	United States of America	95	638,201
2000-09	**Direct Sales**		**96**	**699,403**
2000-09			**99**	**833,224**

CUBE Extension to GROUP BY

In addition to the subtotals generated by the ROLLUP extension, the CUBE extension will generate subtotals for all combinations of the dimensions specified. If "n" is the number of columns listed in the CUBE, there will be 2^n subtotal combinations.

```
SELECT <column_name_1>,
       <column_name_2>,
aggregate_function(<column_name_N>)
FROM <table_name>
[WHERE condition]
GROUP BY CUBE(<column_name_1>,
              <column_name_2>)
```

CUBE Extension to GROUP BY

```
select COUNTRIES.COUNTRY_NAME COUNTRY,
        CHANNELS.CHANNEL_DESC CHANNEL,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     TIMES,
     COUNTRIES,
     CHANNELS
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      TIMES.CALENDAR_MONTH_DESC='2000-09' AND
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by cube(COUNTRIES.COUNTRY_NAME, CHANNELS.CHANNEL_DESC)
order by 1,2
```

COUNTRY	CHANNEL	AVG_SALES$	SALES$
France	Direct Sales	102	1,217,808
France	Internet	121	228,241
France	Partners	108	666,172
France		**105**	**2,112,221**
United States of America	Direct Sales	102	1,217,808
United States of America	Internet	121	228,241
United States of America	Partners	108	666,172
United States of America		**105**	**2,112,221**
	Direct Sales	**102**	**2,435,616**
	Internet	**121**	**456,482**
	Partners	**108**	**1,332,343**
		105	**4,224,441**

CUBE Extension to GROUP BY

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     CUSTOMERS,
     TIMES,
     CHANNELS,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by CUBE(TIMES.CALENDAR_MONTH_DESC,
              CHANNELS.CHANNEL_DESC,
              COUNTRIES.COUNTRY_NAME)
order by 1,2,3
```

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$
2000-09	Direct Sales	France	116	61,202
2000-09	Direct Sales	United States of America	95	638,201
2000-09	**Direct Sales**		**96**	**699,403**
2000-09	Internet	France	148	9,597
2000-09	Internet	United States of America	117	124,224
2000-09	**Internet**		**119**	**133,821**
2000-09		**France**	**120**	**70,799**
2000-09		**United States of America**	**98**	**762,425**
2000-09			99	833,224
	Direct Sales	**France**	**116**	**61,202**
	Direct Sales	**United States of America**	**95**	**638,201**
	Direct Sales		**96**	**699,403**
	Internet	**France**	**148**	**9,597**
	Internet	**United States of America**	**117**	**124,224**
	Internet		**119**	**133,821**
		France	**120**	**70,799**
		United States of America	**98**	**762,425**
			99	**833,224**

Partial CUBE

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     CUSTOMERS,
     TIMES,
     CHANNELS,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by TIMES.CALENDAR_MONTH_DESC,
         CUBE(CHANNELS.CHANNEL_DESC,
              COUNTRIES.COUNTRY_NAME)
order by 1,2,3
```

Partial CUBE does not produce a grand total row.

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$
2000-09	Direct Sales	France	116	61,202
2000-09	Direct Sales	United States of America	95	638,201
2000-09	**Direct Sales**		**96**	**699,403**
2000-09	Internet	France	148	9,597
2000-09	Internet	United States of America	117	124,224
2000-09	**Internet**		**119**	**133,821**
2000-09		**France**	**120**	**70,799**
2000-09		**United States of America**	**98**	**762,425**
2000-09			**99**	**833,224**

GROUPING Functions

Two challenges arise with the use of ROLLUP and CUBE.

✓ First, how can you programmatically determine which result set rows are subtotals, and how do you find the exact level of aggregation for a given subtotal? You often need to use subtotals in calculations such as percent-of-totals, so you need an easy way to determine which rows are the subtotals.

✓ Second, what happens if query results contain both stored NULL values and "NULL" values created by a ROLLUP or CUBE? How can you differentiate between the two?

GROUPING Functions handle these problems.

GROUPING Function

GROUPING function It accepts a single column as a parameter and returns "1" if the column contains a null value generated as part of a subtotal by a ROLLUP or CUBE operation or "0" for any other value, including stored null values.

```
SELECT <column_name_1>,
       <column_name_2>,
aggregate_function(<column_name_N>),
GROUPING(<column_name_1>)
FROM <table_name>
[WHERE condition]
GROUP BY {CUBE|ROLLUP|GROUPING SETS
          (<column_name_1>,
           <column_name_2>)}
```

GROUPING Function

```
select COUNTRIES.COUNTRY_NAME COUNTRY,
       CHANNELS.CHANNEL_DESC CHANNEL,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$,
       grouping(COUNTRIES.COUNTRY_NAME) GR_COUNTRY,
       grouping(CHANNELS.CHANNEL_DESC) GR_CHANNEL
from SALES,
     TIMES,
     COUNTRIES,
     CHANNELS
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      TIMES.CALENDAR_MONTH_DESC='2000-09' AND
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by cube(COUNTRIES.COUNTRY_NAME, CHANNELS.CHANNEL_DESC)
order by 1,2
```

COUNTRY	CHANNEL	AVG_SALES$	SALES$	GR_COUNTRY	GR_CHANNEL
France	Direct Sales	102	1,217,808	0	0
France	Internet	121	228,241	0	0
France	Partners	108	666,172	0	0
France		**105**	**2,112,221**	0	1
United States of America	Direct Sales	102	1,217,808	0	0
United States of America	Internet	121	228,241	0	0
United States of America	Partners	108	666,172	0	0
United States of America		**105**	**2,112,221**	0	1
	Direct Sales	**102**	**2,435,616**	1	0
	Internet	**121**	**456,482**	1	0
	Partners	**108**	**1,332,343**	1	0
		105	**4,224,441**	1	1

GROUPING Function & DECODE

```
select DECODE(grouping(COUNTRIES.COUNTRY_NAME), 1, 'All Countries', COUNTRIES.COUNTRY_NAME) COUNTRY,
       DECODE(grouping(CHANNELS.CHANNEL_DESC), 1, 'All Channels',CHANNELS.CHANNEL_DESC) CHANNEL,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$,
       grouping(COUNTRIES.COUNTRY_NAME) GR_COUNTRY,
       grouping(CHANNELS.CHANNEL_DESC) GR_CHANNEL
from SALES,
     TIMES,
     COUNTRIES,
     CHANNELS
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by rollup(COUNTRIES.COUNTRY_NAME, CHANNELS.CHANNEL_DESC)
order by 5,1
```

COUNTRY	CHANNEL	AVG_SALES$	SALES$	GR_COUNTRY	GR_CHANNEL
France	Internet	121	228,241	0	0
France	Partners	108	666,172	0	0
France	Direct Sales	102	1,217,808	0	0
France	**All Channels**	**105**	**2,112,221**	**0**	**1**
United States of America	Internet	121	228,241	0	0
United States of America	Partners	108	666,172	0	0
United States of America	Direct Sales	102	1,217,808	0	0
United States of America	**All Channels**	**105**	**2,112,221**	**0**	**1**
All Countries	**All Channels**	**105**	**4,224,441**	**1**	**1**

GROUPING_ID Function

The **GROUPING_ID** function provides an alternate and more compact way to identify subtotal rows. Passing the dimension columns as arguments, it returns a number indicating the GROUP BY level.

```
SELECT <column_name_1>,
       <column_name_2>,
aggregate_function(<column_name_N>),
GROUPING_ID(<column_name_1>, <column_name_2>)
FROM <table_name>
[WHERE condition]
GROUP BY {CUBE|ROLLUP|GROUPING SETS
          (<column_name_1>,
           <column_name_2>)}
```

GROUPING_ID Function

For each row, **GROUPING_ID** takes the set of 1's and 0's that would be generated if you used the appropriate GROUPING functions and concatenates them, forming a bit vector. The bit vector is treated as a binary number, and the number's base-10 value is returned by the **GROUPING_ID** function.

GROUPING_ID Example for CUBE(a, b):

Aggregation Level	Bit Vector	GROUPING_ID
a, b	0 0	0
a	0 1	1
b	1 0	2
Grand Total	1 1	3

GROUPING_ID Function

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$,
       GROUPING_ID (TIMES.CALENDAR_MONTH_DESC) GR_TIME_1,
       GROUPING_ID (CHANNELS.CHANNEL_DESC) GR_CHANNEL_1,
       GROUPING_ID (COUNTRIES.COUNTRY_NAME) GR_COUNTRY_1,
       GROUPING_ID (COUNTRIES.COUNTRY_NAME, CHANNELS.CHANNEL_DESC) GR_2,
       GROUPING_ID (TIMES.CALENDAR_MONTH_DESC,CHANNELS.CHANNEL_DESC, COUNTRIES.COUNTRY NAME) GR_3
from SH.SALES,
     SH.CUSTOMERS,
     SH.TIMES,
     SH.CHANNELS,
     SH.COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by CUBE(TIMES.CALENDAR_MONTH_DESC, CHANNELS.CHANNEL_DESC, COUNTRIES.COUNTRY_NAME)
order by 1,2,3
```

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$	GR_TIME_1	GR_CHANNEL_1	GR_COUNTRY_1	GR_2	GR_3
2000-09	Direct Sales	France	116	61,202	0	0	0	0	0
2000-09	Direct Sales	United States of America	95	638,201	0	0	0	0	0
2000-09	Direct Sales		96	699,403	0	0	1	2	1
2000-09	Internet	France	148	9,597	0	0	0	0	0
2000-09	Internet	United States of America	117	124,224	0	0	0	0	0
2000-09	Internet		119	133,821	0	0	1	2	1
2000-09		France	120	70,799	0	1	0	1	2
2000-09		United States of America	98	762,425	0	1	0	1	2
2000-09			99	833,224	0	1	1	3	3
	Direct Sales	France	116	61,202	1	0	0	0	4
	Direct Sales	United States of America	95	638,201	1	0	0	0	4
	Direct Sales		96	699,403	1	0	1	2	5
	Internet	France	148	9,597	1	0	0	0	4
	Internet	United States of America	117	124,224	1	0	0	0	4
	Internet		119	133,821	1	0	1	2	5
		France	120	70,799	1	1	0	1	6
		United States of America	98	762,425	1	1	0	1	6
			99	833,224	1	1	1	3	7

GROUPING_ID Function & HAVING

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$,
       GROUPING_ID (TIMES.CALENDAR_MONTH_DESC) GR_TIME_1,
       GROUPING_ID (CHANNELS.CHANNEL_DESC) GR_CHANNEL_1,
       GROUPING_ID (COUNTRIES.COUNTRY_NAME) GR_COUNTRY_1,
       GROUPING_ID (COUNTRIES.COUNTRY_NAME, CHANNELS.CHANNEL_DESC) GR_2,
       GROUPING_ID (TIMES.CALENDAR_MONTH_DESC,CHANNELS.CHANNEL_DESC, COUNTRIES.COUNTRY_NAME) GR_3
from SH.SALES,
     SH.CUSTOMERS,
     SH.TIMES,
     SH.CHANNELS,
     SH.COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by cube(TIMES.CALENDAR_MONTH_DESC, CHANNELS.CHANNEL_DESC, COUNTRIES.COUNTRY_NAME)
having GROUPING_ID (TIMES.CALENDAR_MONTH_DESC,CHANNELS.CHANNEL_DESC, COUNTRIES.COUNTRY_NAME) > 0
order by 1,2,3
```

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$	GR_TIME_1	GR_CHANNEL_1	GR_COUNTRY_1	GR_2	GR_3
2000-09	Direct Sales		96	699,403	0	0	1	2	1
2000-09	Internet		119	133,821	0	0	1	2	1
2000-09		France	120	70,799	0	1	0	1	2
2000-09		United States of America	98	762,425	0	1	0	1	2
2000-09			99	833,224	0	1	1	3	3
	Direct Sales	France	116	61,202	1	0	0	0	4
	Direct Sales	United States of America	95	638,201	1	0	0	0	4
	Direct Sales		96	699,403	1	0	1	2	5
	Internet	France	148	9,597	1	0	0	0	4
	Internet	United States of America	117	124,224	1	0	0	0	4
	Internet		119	133,821	1	0	1	2	5
		France	120	70,799	1	1	0	1	6
		United States of America	98	762,425	1	1	0	1	6
			99	833,224	1	1	1	3	7

GROUP_ID Function

It's possible to write queries that return the duplicate subtotals, which can be a little confusing. The **GROUP_ID** function assigns the value "0" to the first set, and all subsequent sets get assigned a higher number. The following query forces duplicates to show the **GROUP_ID** function in action.

```
SELECT <column_name_1>,
       <column_name_2>,
aggregate_function(<column_name_N>),
GROUP_ID()
FROM <table_name>
[WHERE condition]
GROUP BY {CUBE|ROLLUP|GROUPING SETS
          (<column_name_1>,
           <column_name_2>)}
```

GROUP_ID Function

```
select COUNTRIES.COUNTRY_NAME COUNTRY,
       CHANNELS.CHANNEL_DESC CHANNEL,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$,
       GROUP_ID()
from SALES,
     TIMES,
     COUNTRIES,
     CHANNELS
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      TIMES.CALENDAR_MONTH_DESC='2000-09' AND
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by COUNTRIES.COUNTRY_NAME, ROLLUP(COUNTRIES.COUNTRY_NAME, CHANNELS.CHANNEL_DESC)
ORDER BY 1,2
```

COUNTRY	CHANNEL	AVG_SALES$	SALES$	GROUP_ID()
France	Direct Sales	102	1,217,808	0
France	Internet	121	228,241	0
France	Partners	108	666,172	0
France		105	2,112,221	0
France		105	2,112,221	1
United States of America	Direct Sales	102	1,217,808	0
United States of America	Internet	121	228,241	0
United States of America	Partners	108	666,172	0
United States of America		105	2,112,221	0
United States of America		105	2,112,221	1

GROUP_ID Function & HAVING

```
select COUNTRIES.COUNTRY_NAME COUNTRY,
       CHANNELS.CHANNEL_DESC CHANNEL,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$,
       GROUP_ID()
from SALES,
     TIMES,
     COUNTRIES,
     CHANNELS
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      TIMES.CALENDAR_MONTH_DESC='2000-09' AND
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by COUNTRIES.COUNTRY_NAME, rollup(COUNTRIES.COUNTRY_NAME, CHANNELS.CHANNEL_DESC)
HAVING GROUP_ID() = 0
ORDER BY 1,2
```

COUNTRY	CHANNEL	AVG_SALES$	SALES$	GROUP_ID()
France	Direct Sales	102	1,217,808	0
France	Internet	121	228,241	0
France	Partners	108	666,172	0
France		105	2,112,221	0
United States of America	Direct Sales	102	1,217,808	0
United States of America	Internet	121	228,241	0
United States of America	Partners	108	666,172	0
United States of America		105	2,112,221	0

GROUPING SETS Expression

You can selectively specify the set of groups that you want to create using a **GROUPING SETS** expression within a GROUP BY clause. This allows precise specification across multiple dimensions without computing the whole CUBE.

```
SELECT <column_name_1>,
       <column_name_2>,
aggregate_function(<column_name_N>) FROM
<table_name>
[WHERE condition]
GROUP BY GROUPING SETS
          (<column_name_1>,
           <column_name_2>)
```

GROUPING SETS Expression

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     CUSTOMERS,
     TIMES,
     CHANNELS,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by grouping sets(TIMES.CALENDAR_MONTH_DESC,
                       CHANNELS.CHANNEL_DESC,
                       COUNTRIES.COUNTRY_NAME)
order by 1,2,3
```

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$
2000-09			99	833,224
	Direct Sales		96	699,403
	Internet		119	133,821
		France	120	70,799
		United States of America	98	762,425

Composite Columns

Composite columns allow columns to be grouped together with braces so they are treated as a single unit when determining the necessary groupings. In the following ROLLUP columns "a" and "b" have been turned into a composite column by the additional braces. As a result the group of "a" is not longer calculated as the column "a" is only present as part of the composite column in the statement.

```
ROLLUP ((a, b), c)
(a, b, c)
(a, b)
()

Not considered:
(a)
```

In a similar way, the possible combinations of the following CUBE are reduced because references to "a" or "b" individually are not considered as they are treated as a single column when the groupings are determined.

```
CUBE ((a, b), c)
(a, b, c)
(a, b)
(c)
()

Not considered:
(a, c)
(a)
(b, c)
(b)
```

Composite Columns

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     CUSTOMERS,
     TIMES,
     CHANNELS,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by grouping sets((TIMES.CALENDAR_MONTH_DESC,CHANNELS.CHANNEL_DESC),
                       COUNTRIES.COUNTRY_NAME)
order by 1,2,3
```

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$
2000-09	Direct Sales		96	699,403
2000-09	Internet		119	133,821
		France	120	70,799
		United States of America	98	762,425

Composite Columns

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     CUSTOMERS,
     TIMES,
     CHANNELS,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by grouping sets((TIMES.CALENDAR_MONTH_DESC,CHANNELS.CHANNEL_DESC,
                        COUNTRIES.COUNTRY_NAME))
order by 1,2,3
```

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$
2000-09	Direct Sales	France	116	61,202
2000-09	Direct Sales	United States of America	95	638,201
2000-09	Internet	France	148	9,597
2000-09	Internet	United States of America	117	124,224

Concatenated Groupings

Concatenated groupings are defined by putting together multiple GROUPING SETS, CUBEs or ROLLUPs separated by commas. The resulting groupings are the cross-product of all the groups produced by the individual grouping sets.

```
GROUPING SETS(a, b), GROUPING SETS(c, d)
(a, c)
(a, d)
(b, c)
(b, d)
```

Concatenated Groupings

```
select TIMES.CALENDAR_MONTH_DESC YEAR_MONTH,
       CHANNELS.CHANNEL_DESC CHANNEL,
       COUNTRIES.COUNTRY_NAME COUNTRY,
       TO_CHAR(AVG(AMOUNT_SOLD), '9,999,999,999') AVG_SALES$,
       TO_CHAR(SUM(AMOUNT_SOLD), '9,999,999,999') SALES$
from SALES,
     CUSTOMERS,
     TIMES,
     CHANNELS,
     COUNTRIES
where SALES.TIME_ID=TIMES.TIME_ID and
      SALES.CUST_ID=CUSTOMERS.CUST_ID and
      SALES.CHANNEL_ID= CHANNELS.CHANNEL_ID and
      CHANNELS.CHANNEL_DESC in ('Direct Sales', 'Internet') and
      TIMES.CALENDAR_MONTH_DESC='2000-09' and
      CUSTOMERS.COUNTRY_ID=COUNTRIES.COUNTRY_ID and
      COUNTRIES.COUNTRY_ISO_CODE in ('US','FR')
group by grouping sets(TIMES.CALENDAR_MONTH_DESC,CHANNELS.CHANNEL_DESC),
         grouping sets (COUNTRIES.COUNTRY_NAME)
```

YEAR_MONTH	CHANNEL	COUNTRY	AVG_SALES$	SALES$
2000-09		France	120	70,799
2000-09		United States of America	98	762,425
	Direct Sales	United States of America	95	638,201
	Internet	United States of America	117	124,224
	Internet	France	148	9,597
	Direct Sales	France	116	61,202

Oracle SQL for Aggregation in Data Warehouses

Q&A

MTN.BI.03

Oracle SQL for Aggregation in Data Warehouses

Author: Volha Kutsevol
Lead Software Engineer
Volha_Kutsevol@epam.com