Extension to Group By

- Group By does not generate totals or sub totals
- CUBE, ROLLUP and GROUPING SETS are extensions to the Group By clause that will allow you to generate subtotals

STORE ID	LOCATION ID	SALES
10	1	100,000
20	1	20,000
30	1	80,000
40	2	50,000
50	2	50,000

STORE ID	LOCATION ID	SALES
10	1	100,000
20	1	20,000
30	1	80,000
-	1	200,000
40	2	50,000
50	2	50,000
-	2	100,000
-	-	300,000

Rollup

- The ROLLUP extension allows you to generate subtotals
- The ROLLUP extension groups totals based on the input hierarchy

```
SELECT
COLUMN1,
COLUMN2,
SUM(COLUMN3)
FROM TABLE
GROUP BY ROLLUP(COLUMN1, COLUMN2);
```

(COLUMN1, COLUMN2)

(COLUMN1)

()

Rollup

```
SELECT
COLUMN1,
COLUMN2,
COLUMN3,
SUM(COLUMN3)
FROM TABLE
GROUP BY ROLLUP(COLUMN1, COLUMN2, COLUMN3);
```

(COLUMN1, COLUMN2, COLUMN3)

(COLUMN1, COLUMN2)

(COLUMN1)

()

Cube

- The CUBE extension allows you to generate subtotals
- The CUBE extension groups totals based on all combinations of the specified columns

```
SELECT
COLUMN1,
COLUMN2,
SUM(COLUMN3)
FROM TABLE
GROUP BY ROLLUP(COLUMN1, COLUMN2);
```

(COLUMN1, COLUMN2)

(COLUMN1)

(COLUMN2)

()

Cube

```
SELECT
COLUMN1,
COLUMN2,
COLUMN3,
SUM(COLUMN3)
FROM TABLE
GROUP BY ROLLUP(COLUMN1, COLUMN2, COLUMN3);
```

(COLUMN1, COLUMN2, COLUMN3)

(COLUMN1, COLUMN2)

(COLUMN2, COLUMN3)

(COLUMN1, COLUMN3)

(COLUMN3)

(COLUMN2)

(COLUMN1)

()

Grouping Sets

- Instead of using CUBE and ROLLUP you can specify your own grouping set instead of evaluating the combinations from the previous extensions

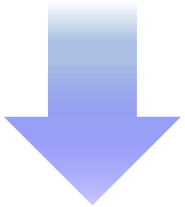
```
SELECT
COLUMN1,
COLUMN2,
SUM(COLUMN3)
FROM TABLE
GROUP BY
GROUPING SET ((COLUMN1, COLUMN2),COLUMN2);
```

(COLUMN1, COLUMN2)

(COLUMN2)

Grouping Sets

GROUPING SET ((COLUMN3, COLUMN2), (COLUMN2, COLUMN1), COLUMN1)



(COLUMN3, COLUMN2)

(COLUMN2, COLUMN1)

(COLUMN1)

GROUPING_ID function

- Allows you to identify which row is a sub-total or grand total, or if the row is neither
    - It will return flag values to represent subtotals, totals and grouping totals
- Only applicable in SELECT statement when used with Group By extensions

```
SELECT
COLUMN1,
COLUMN2,
GROUPING_ID(COLUMN1, COLUMN2) AS GROUP_ID,
SUM(COLUMN3)
FROM TABLE
GROUP BY ROLLUP(COLUMN1, COLUMN2);
```

GROUPING_ID function

GROUPING_ID(COLUMN1, COLUMN2)

GROUP_ID = 0 (COLUMN1, COLUMN2)

GROUP_ID = 1 (COLUMN1)

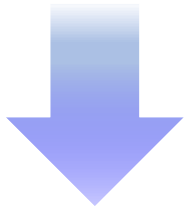
GROUP_ID = 2 (COLUMN2)

GROUP_ID = 3 ()

Composite columns

- In your ROLLUP and CUBE extensions you can use composite columns to return subtotals for different combinations

ROLLUP (COLUMN1, COLUMN2, COLUMN3)



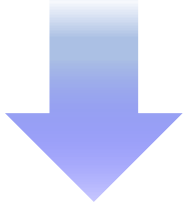
(COLUMN1, COLUMN2, COLUMN3)

(COLUMN1,COLUMN2)

(COLUMN1)

()

ROLLUP ((COLUMN1, COLUMN2),COLUMN2)



(COLUMN1, COLUMN2, COLUMN3)

(COLUMN1,COLUMN2)

()