

0. GROUP BY 절

```
SELECT 상품ID, 월, SUM(매출액) AS 매출액
FROM 월별매출
GROUP BY 상품ID, 월
```

상품ID	월	매출액
1 P002	2019.10	10000
2 P003	2019.11	10000
3 P002	2019.11	20000
4 P003	2019.10	15000
5 P001	2019.10	15000
6 P001	2019.11	25000

가장 기본적이고 단순한 GROUP BY절만 사용한 결과입니다. 상품ID와 월에 대해 매출액의 합계를 구해주었습니다.

1. ROLLUP

ROLLUP함수는 소그룹간의 합계를 계산하는 함수입니다. ROLLUP을 사용하면 GROUP BY로 묶은 각각의 소그룹 합계와 전체 합계를 모두 구할 수 있습니다.

```
SELECT 상품ID, 월, SUM(매출액) AS 매출액
FROM 월별매출
GROUP BY ROLLUP(상품ID, 월);
```

상품ID	월	매출액	
1 P001	2019.10	15000	
2 P001	2019.11	25000	
3 P001	(null)	40000	→ P001 합계
4 P002	2019.10	10000	
5 P002	2019.11	20000	
6 P002	(null)	30000	→ P002 합계
7 P003	2019.10	15000	
8 P003	2019.11	10000	
9 P003	(null)	25000	→ P003 합계
10 (null)	(null)	95000	→ 전체 합계

NULL값으로 표시된 부분들이 바로 ROLLUP함수를 써서 나온 합계입니다.

P001 그룹의 매출액 합계, P002 그룹의 매출액 합계, P003 그룹의 매출액 합계가 각각 계산되었으며 전체 Total 합계 또한 한 번에 표시할 수 있습니다.

결과값에서 알 수 있듯이 GROUP BY절에 명시한 모든 컬럼에 대한 소그룹의 합계를 구해주는 것이 아니고 맨 처음 명시한 컬럼에 대해서만 소그룹 합계를 구해줍니다.

이때 각 그룹명은 따로 지정하지 않았기 때문에 null값으로 표시되었습니다. 그룹명을 지정하는 것은 맨 마지막 GROUPING 함수에서 자세히 다루겠습니다.

GROUP BY ROLLUP(컬럼1, 컬럼2)

```
=
GROUP BY 컬럼1, 컬럼2
UNION ALL
GROUP BY 컬럼1
UNION ALL
모든 집합 그룹 결과
```

2. CUBE

CUBE함수는 항목들 간의 다차원적인 소계를 계산합니다. ROLLUP과 달리 GROUP BY절에 명시한 모든 컬럼에 대해 소그룹 합계를 계산해줍니다.

```
SELECT 상품ID, 월, SUM(매출액) AS 매출액
FROM 월별매출
GROUP BY CUBE(상품ID, 월);
```

상품ID	월	매출액	
1 (null)	(null)	95000	→ 전체 합계
2 (null)	2019.10	40000	→ 2019.10 합계
3 (null)	2019.11	55000	→ 2019.11 합계
4 P001	(null)	40000	→ P001 합계
5 P001	2019.10	15000	
6 P001	2019.11	25000	
7 P002	(null)	30000	→ P002 합계
8 P002	2019.10	10000	
9 P002	2019.11	20000	
10 P003	(null)	25000	→ P003 합계
11 P003	2019.10	15000	
12 P003	2019.11	10000	

ROLLUP함수를 사용했을 때보다 결과가 좀 더 복잡합니다. 상품ID별 합계뿐만 아니라 월별 합계까지 한 번에 볼 수 있습니다.

GROUP BY CUBE(컬럼1, 컬럼2)

```
=
GROUP BY 컬럼1, 컬럼2
UNION ALL
GROUP BY 컬럼1
UNION ALL
GROUP BY 컬럼2
UNION ALL
모든 집합 그룹 결과
```

3. GROUPING SETS

GROUPING SETS는 특정 항목에 대한 소계를 계산하는 함수입니다.

```
SELECT 상품ID, 월, SUM(매출액) AS 매출액
FROM 월별매출
GROUP BY GROUPING SETS(상품ID, 월);
```

상품ID	월	매출액	
1 P001	(null)	40000	→ 상품별 합계
2 P003	(null)	25000	
3 P002	(null)	30000	
4 (null)	2019.11	55000	→ 월별 합계
5 (null)	2019.10	40000	

앞의 ROLLUP, CUBE에 비해 훨씬 결과가 단순합니다.

ROLLUP과 CUBE는 GROUP BY 결과에 소그룹 합계와 토털 합계를 보여주지만 GROUPING SETS는 각 소그룹별 합계만 간단하게 보여줍니다.

GROUPING SETS 함수는 각각의 컬럼으로 GROUP BY한 값을 UNION ALL 한 것과 동일한 결과를 보여줍니다.

```
SELECT 상품ID, 월, SUM(매출액) AS 매출액
FROM 월별매출
GROUP BY 상품ID
UNION ALL
SELECT 상품ID, 월, SUM(매출액) AS 매출액
FROM 월별매출
GROUP BY 월
```

4. GROUPING

GROUPING은 직접적으로 그룹별 집계를 구하는 함수는 아니지만 위의 집계함수들을 지원하는 함수입니다.

집계가 계산된 결과에 대해서는 1의 값을 갖고 그렇지 않은 결과에 대해서는 0의 값을 갖습니다.

```
SELECT
CASE GROUPING(상품ID) WHEN 1 THEN '모든 상품ID' ELSE 상품ID END AS 상품ID,
CASE GROUPING(월) WHEN 1 THEN '모든 월' ELSE 월 END AS 월,
SUM(매출액) AS 매출액
FROM 월별매출
GROUP BY ROLLUP(상품ID, 월);
```

상품ID	월	매출액
1 P001	2019.10	15000
2 P001	2019.11	25000
3 P001	모든 월	40000
4 P002	2019.10	10000
5 P002	2019.11	20000
6 P002	모든 월	30000
7 P003	2019.10	15000
8 P003	2019.11	10000
9 P003	모든 월	25000
10 모든 상품ID	모든 월	95000

CASE WHEN문을 사용해서 맨 처음에 단순 ROLLUP함수만 썼을 때 NULL값으로 표시되었던 곳에 값을 넣어주었습니다.

집계가 계산된 결과에 대해서만 값을 넣어주면 되기 때문에 GROUPING(컬럼명)=1인 경우에만 '모든상품ID' 또는 '모든월' 값을 부여했고 0인 경우에는 원래대로 상품ID와 월을 써주었습니다.