

집계함수와 group by

강사 : 강병준

Group 함수

- 그룹 함수란 여러 행 또는 테이블 전체의 행에 대해 함수가 적용되어 하나의 결과값을 가져오는 함수를 말함
- **GROUP BY**절을 이용하여 그룹 당 하나의 결과가 주어지도록 그룹화 할 수 있다
- **HAVING**절을 사용하여 그룹 함수를 가지고 조건비교를 할 수 있음
- COUNT(*)를 제외한 모든 그룹함수는 NULL값을 고려하지 않음
- **MIN, MAX** 그룹함수는 모든 자료형에 대해서 사용 할 수 있음

Group 함수 란?

- Group functions operate on sets of rows to give one result per group.

EMP

DEPTNO	SAL
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

“maximum
salary in
the EMP table”

MAX (SAL)

5000

Group 함수

```
SELECT      [column,] group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY   column]
[ORDER BY   column] ;
```

- AVG
- COUNT
- MAX
- MIN
- SUM
- ...
- ** See SQL Reference Books

Group 함수의 종류1

☞ **COUNT** : COUNT 함수는 검색된 행의 수를 반환

```
SQL>SELECT COUNT(deptno) FROM DEPT ;
```

```
COUNT(DEPTNO)
```

```
-----
```

```
4
```

==> 검색된 행의 총 수 4개를 반환
즉 4개의 부서가 존재

☞ **MAX** : MAX 함수는 컬럼중의 최대값을 반환

```
SQL>SELECT MAX(sal) salary FROM emp ;
```

```
SALARY
```

```
-----
```

```
5000
```

==>

sal 컬럼중에서 제일 큰값을 반환
즉 가장 큰 급여를 반환

Group 함수의 종류2

☞ **MIN** : MIN 함수는 컬럼중의 최소값을 반환

```
SQL>SELECT MIN(sal) salary FROM emp ;  
SALARY
```

800 ==> sal 컬럼중에서 가장 작은 값 반환
즉 가장 적은 급여를 반환

☞ **AVG** : AVG 함수는 평균값을 반환

```
SQL>SELECT ROUND(AVG(sal),1) salary  
FROM emp WHERE deptno = 30  
SALARY
```

1566.7 ==> 30부서 사원의 평균 급여를
소수점 1자리 이하에서 반올림해서 보여줌

Group 함수의 종류3

☞ **SUM** : SUM 함수는 검색된 컬럼의 합을 반환

```
SQL>SELECT SUM(sal) salary FROM emp
      WHERE deptno = 30;
      SALARY
```

9400

==> 30부서 사원의 급여 합계를 보여줌

☞ **STDDEV** : STDDEV 함수는 표준편차를 반환

```
SQL> SELECT ROUND(STDDEV(sal),3) salary
      FROM emp WHERE deptno = 30 ;
      SALARY
```

668.331

==> 30부서 사원의 급여 표준편차를 반환

GROUP BY 기본 사항

- GROUP BY 절 사용

- 특정한 컬럼의 데이터 들을 다른 데이터들과 비교해 유일한 값에 따라 그룹을 만듦
- GROUP BY절을 사용하여 한 테이블의 행들을 원하는 그룹으로 나눈다
- Column명을 GROUP함수와 SELECT절에 사용하고자 하는 경우 GROUP BY뒤에 Column명을 추가

GROUP BY 기본 구문

```
SELECT      column, group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[ORDER BY   column];
```

- Divide rows in a table into smaller groups by using the GROUP BY clause.

Using the GROUP BY

EMP

DEPTNO	SAL
-----	-----
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

2916.6667

2175

1566.6667

“average
salary
in EMP
table
for each
department”

DEPTNO	AVG (SAL)
-----	-----
10	2916.6667
20	2175
30	1566.6667

Using the GROUP BY

- All columns in the SELECT list that are not in group functions must be in the GROUP BY clause.

```
SQL> SELECT deptno, AVG(sal)
2 FROM emp
3 GROUP BY deptno;
```

DEPTNO	AVG (SAL)
10	2916.6667
20	2175
30	1566.6667

Using the GROUP BY

- The GROUP BY column does not have to be in the SELECT list.

```
SQL> SELECT      AVG(sal)
2  FROM          emp
3  GROUP BY deptno;
```

```
AVG(SAL)
-----
2916.6667
2175
1566.6667
```

Grouping by More than One Column

EMP

DEPTNO	JOB	SAL
10	MANAGER	2450
10	PRESIDENT	5000
10	CLERK	1300
20	CLERK	800
20	CLERK	1100
20	ANALYST	3000
20	ANALYST	3000
20	MANAGER	2975
30	SALESMAN	1600
30	MANAGER	2850
30	SALESMAN	1250
30	CLERK	950
30	SALESMAN	1500
30	SALESMAN	1250

“sum salaries in
the EMP table
for each job,
grouped by
department”

DEPTNO	JOB	SUM (SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	6000
20	CLERK	1900
20	MANAGER	2975
30	CLERK	950
30	MANAGER	2850
30	SALESMAN	5600

Why error ?

```
SQL> SELECT    deptno, COUNT(ename)
      2  FROM      emp;
```

```
SELECT deptno, COUNT(ename)
      *
```

Why error??

ERROR at line 1:

ORA-00937: not a single-group group function

```
SQL> SELECT    deptno, AVG(sal)
      2  FROM      emp
      3  WHERE      AVG(sal) > 2000
      4  GROUP BY  deptno;
```

```
WHERE AVG(sal) > 2000
      *
```

Why error??

ERROR at line 3:

ORA-00934: group function is not allowed here

EMP 테이블에서 모든 SALESMAN 에 대하여 급여의 평균, 최고액, 최저액, 합계를 구하여 출력하여라...

```
SQL> SELECT AVG(sal), MAX(sal), MIN(sal), SUM(sal),  
2 FROM emp,  
3 WHERE job LIKE 'SAL%';
```

```
..  
AVG(SAL)  MAX(SAL)  MIN(SAL)  SUM(SAL),  
-----  
1400      1600      1250      5600,
```

아래의 SELECT 문장을 분석하여라...

```
SQL> SELECT MIN(ename),MAX(ename),MIN(hiredate),  
2 MAX(hiredate),MIN(sal),MAX(sal),  
3 FROM emp;
```

```
..  
MIN(ENAME) MAX(ENAME) MIN(HIREDATE)  MAX(HIREDATE)  MIN(SAL)  MAX(SAL),  
-----  
ADAMS      WARD      17-DEC-80      12-JAN-83      800      5000,
```

EMP 테이블에 등록되어 있는 인원수, 보너스에 NULL 이 아닌 인원수, 보너스의 평균, 등록되어 있는 부서의 수를 구하여 출력하여라...

```
SQL> SELECT COUNT(*) c_inwon, COUNT(comm) c_comm, AVG(comm) a_comm,
2      AVG(NVL(comm,0)) n_comm, COUNT(deptno) c_dept,
3      COUNT(DISTINCT deptno) c_dis,
4      FROM emp;
```

C_INWON	C_COMM	A_COMM	N_COMM	C_DEPT	C_DIS
14	4	550	157.14286	14	3

데이터 그룹 생성

지금까지 모든 그룹 함수는 테이블을 하나의 큰 그룹으로 다루었다. 여기서는 테이블의 자료를 보다 작은 그룹으로 나누어 사용할 수 있는 GROUP BY 절을 배워보자...

Syntax

```
SELECT      [column,] group_function(column) [,group_function(column),...]
FROM        table_name
[WHERE      condition]
[GROUP BY  group_by_expression]
[HAVING     condition]
[ORDER BY  column];
```

table_name 테이블명 질의 대상 테이블 이름..

WHERE condition 을 만족하는 행들만 검색..

ORDER BY	질의 결과 정렬을 위한 옵션(ASC:오름차순(Default),DESC 내림차순)..
group_function	Group Function이 SELECT 절 뒤에서 Column 과 같이.. 기술되면 반드시 GROUP BY 절이 기술되어야 한다..
GROUP BY group_by_expression	행을 그룹하기 위한 기준이 될 Column 을 기술..
HAVING condition	그룹에 대한 조건을 기술..

SELECT 절..

SELECT 절에 GROUP 함수와 Column 이 같이 기술되면 반드시 GROUP BY 절이 기술되어야 한다. 그러나 SELECT 절에 GROUP 함수만 기술되고 Column 은 기술되지 않으면 반드시 GROUP BY 절을 기술할 필요는 없다...

```
SQL> SELECT deptno,COUNT(*),AVG(sal),MIN(sal),MAX(sal),SUM(sal),
2 FROM emp;
SELECT deptno,COUNT(*),AVG(sal),MIN(sal),MAX(sal),SUM(sal),
*,
ERROR at line 1:
ORA-00937: not a single-group group function.
```

EMP 테이블에서 부서별로 인원수, 평균 급여, 최저급여, 최고 급여, 급여의 합을 구하여 출력하여라...

```
SQL> SELECT deptno,COUNT(*),AVG(sal),MIN(sal),MAX(sal),SUM(sal)..  
2 FROM emp..  
3 GROUP BY deptno;..
```

DEPTNO	COUNT(*)	AVG(SAL)	MIN(SAL)	MAX(SAL)	SUM(SAL)
10	3	2916.6667	1300	5000	8750
20	5	2175	800	3000	10875
30	6	1566.6667	950	2850	9400

각 부서별로 인원수, 급여의 평균, 최저 급여, 최고 급여, 급여의 합을 구하여 급여의 합이 많은 순으로 출력하여라...

```
SQL> SELECT deptno,COUNT(*),AVG(sal),MIN(sal),MAX(sal),SUM(sal)..  
2 FROM emp..  
3 GROUP BY deptno..  
4 ORDER BY SUM(sal) DESC;..
```

DEPTNO	COUNT(*)	AVG(SAL)	MIN(SAL)	MAX(SAL)	SUM(SAL)
20	5	2175	800	3000	10875
30	6	1566.6667	950	2850	9400
10	3	2916.6667	1300	5000	8750

각 부서별로 인원수, 급여의 평균, 최저 급여, 최고 급여, 급여의 합을 구하여 급여의 합이 많은 순으로 출력하여라...

```
SQL> SELECT COUNT(*),AVG(sal),MIN(sal),MAX(sal),SUM(sal)
2 FROM emp
3 GROUP BY deptno
4 ORDER BY SUM(sal) DESC;
```

COUNT(*)	AVG(SAL)	MIN(SAL)	MAX(SAL)	SUM(SAL)
5	2175	800	3000	10875
6	1566.6667	950	2850	9400
3	2916.6667	1300	5000	8750

❖ 참고

위 결과는 어느 행이 어떤 부서의 결과인지 알 수 없다. 즉 GROUP BY 절에 기술된 Column 은 SELECT 절에 반드시 기술될 필요는 없다. 그러나 결과를 구분하기가 어렵다...

☞ Guidelines ..

- 1) SELECT 절에 GROUP Function이 포함된다면 GROUP BY 절에 각각의 열이 명시되어야 함..
- 2) WHERE 절을 사용하여 행을 그룹으로 나누기 전에 행을 제외..
- 3) 그룹에 대한 조건은 HAVING 절을 사용(그룹에 대한 조건을 WHERE 절에서 기술 불가)..
- 4) GROUP BY 절에 열을 포함(열의 별칭은 사용할 수 없다)..
- 5) Default 는 GROUP BY 절 다음에 기술된 순서로 오름차순으로 정렬되지만 ORDER BY 절을 이용하여 변경 가능..

1.3.3 하나 이상의 Column 으로 그룹화..

때로는 그룹 내의 그룹에 대한 결과를 검색할 필요가 있다. 이러한 경우 대그룹, 중그룹, 소그룹으로 나누고자 하는 순서대로 GROUP BY 절 뒤에 기술하면 순서대로 오름차순으로 정렬된다...

가) Syntax..

SELECT	[column,] <u>group_function</u> (column) [,group_function(column),...]
FROM	table_name..
[WHERE	condition]..
[GROUP BY	<u>group_by_expr1</u> [,group_by_expr2,]
[HAVING	condition]..
[ORDER BY	column];..

GROUP BY group_by_expr1 [,group_by_expr2,] 행을 그룹하기 위한 기준이 될 Column 을 기술한다...

부서별, 업무별 그룹하여 결과를 부서번호, 업무, 인원수, 급여의 평균, 급여의 합을 구하여 출력하여라.

```
SQL> SELECT deptno,job,COUNT(*),AVG(sal),SUM(sal) FROM emp GROUP BY deptno,job;
```

DEPTNO	JOB	COUNT(*)	AVG(SAL)	SUM(SAL)
10	CLERK	1	1300	1300
10	MANAGER	1	2450	2450
10	PRESIDENT	1	5000	5000

업무별, 부서별 그룹하여 결과를 부서번호, 업무, 인원수, 급여의 평균, 급여의 합을 구하여 출력하여라.

```
SQL> SELECT job,deptno,COUNT(*),AVG(sal),SUM(sal) FROM emp  
GROUP BY job,deptno;
```

JOB	DEPTNO	COUNT(*)	AVG(SAL)	SUM(SAL)
ANALYST	20	2	3000	6000
CLERK	10	1	1300	1300
CLERK	20	2	950	1900
CLERK	30	1	950	950
MANAGER	10	1	2450	2450
MANAGER	20	1	2975	2975
MANAGER	30	1	2850	2850

.....

Having 절 사용

- Having 절과 함께 GROUP BY 절 사용
 - WHERE절에 GROUP Function을 사용할 수 없음
 - HAVING절은 GRUOP 함수를 가지고 조건비교를 할 때 사용
 - WHERE → GRUOP BY → HAVING → ORDER BY순으로 쿼리문이 와야 함

```
SELECT      column, group_function
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   column];
```

Having 절 사용

EMP

DEPTNO	SAL
-----	-----
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

5000

3000

2850

“maximum
salary
per department
greater than
\$2900”

DEPTNO	MAX (SAL)
-----	-----
10	5000
20	3000

Having 절 사용

```
SQL> SELECT      deptno, max(sal)
  2  FROM          emp
  3  GROUP BY      deptno
  4  HAVING        max(sal)>2900;
```

DEPTNO	MAX (SAL)
10	5000
20	3000

Having절 예제

- 예제1 (scott.emp)

조건>

사원수가 5명 이상 넘는 부서의 부서명과 사원수를 출력하시오.

DNAME	COUNT(A.EMPNO)
-----	-----
RESEARCH	5
SALES	6

EMP 테이블에서 급여가 최대 2900 이상인 부서에 대해서 부서번호, 평균 급여, 급여의 합을 구하여 출력하여라...

```
SQL> SELECT deptno, AVG(sal), SUM(sal),  
2 FROM emp,  
3 GROUP BY deptno,  
4 HAVING MAX(sal) > 2900;..
```

```
..  
DEPTNO  AVG(SAL)  SUM(SAL)..  
-----  
10 2916.6667      8750..  
20      2175     10875..
```

EMP 테이블에서 업무별 급여의 평균이 3000 이상인 업무에 대해서 업무명, 평균 급여, 급여의 합을 구하여 출력하여라...

```
SQL> SELECT job, AVG(sal), SUM(sal),  
2 FROM emp,  
3 GROUP BY job,  
4 HAVING AVG(sal) >= 3000;..
```

```
..  
JOB      AVG(SAL)  SUM(SAL)..  
-----  
ANALYST      3000      6000..  
PRESIDENT    5000      5000..
```

EMP 테이블에서 전체 월급이 5000 을 초과하는 각 업무에 대해서 업무와 월급여 합계를 출력하여라. 단 판매원은 제외하고 월 급여 합계로 정렬(내림차순)하여라...

```
SQL> SELECT job,SUM(sal) PAYROLL,  
2 FROM emp..  
3 WHERE job NOT LIKE 'SALES'..  
4 GROUP BY job..  
5 HAVING SUM(sal) > 5000..  
6 ORDER BY SUM(sal) DESC;..
```

JOB	PAYROLL..
MANAGER	8275..
ANALYST	6000..

Guidelines ..

- 1) HAVING 절은 GROUP BY 절 앞에 기술 가능하지만 GROUP BY 절 다음에 기술하는 것이 논리적이므로 권장 됩니다. HAVING 절이 SELECT 절에 있는 그룹에 적용되기 전에 그룹은 구성되고 그룹 함수는 구성됩니다...
- 2) SELECT 절에 그룹 함수를 사용하지 않고 GROUP BY 절만 사용 가능하다. 그룹 함수의 결과로 행이 제한 된다면 HAVING 절 뿐만 아니라 GROUP BY 절을 사용해야 된다...

중첩 그룹 함수

그룹 함수는 어떤 LEVEL까지도 중첩할 수 있습니다.

문제13) 부서별 평균 중 최대 평균 급여, 부서별 급여의 합 중 최대 급여, 부서별 급여의 최소 급여, 부서별 급여의 최대 급여를 출력하여라.

```
SQL> SELECT MAX(AVG(sal)),MAX(SUM(sal)),MIN(MIN(sal)),MAX(MAX(sal))  
2  FROM emp  
3  GROUP BY deptno;
```

MAX(AVG(SAL))	MAX(SUM(SAL))	MIN(MIN(SAL))	MAX(MAX(SAL))
2916.6667	10875	800	5000

◆ 연습문제 ◆

1. EMP 테이블에서 인원수, 최대 급여, 최소 급여, 급여의 합을 계산하여 출력하는 SELECT 문장을 작성하여라.
2. EMP 테이블에서 각 업무별로 최대 급여, 최소 급여, 급여의 합을 출력하는 SELECT 문장을 작성하여라.
3. EMP 테이블에서 업무별 인원수를 구하여 출력하는 SELECT 문장을 작성하여라.
4. EMP 테이블에서 최고 급여와 최소 급여의 차이는 얼마인가 출력하는 SELECT 문장을 작성하여라.
5. EMP 테이블에서 아래의 결과를 출력하는 SELECT 문장을 작성하여라.

H_YEAR	COUNT(*)	MIN(SAL)	MAX(SAL)	AVG(SAL)	SUM(SAL)
80	1	800	800	800	800
81	10	950	5000	2282.5	22825
82	2	1300	3000	2150	4300
83	1	1100	1100	1100	1100

6. EMP 테이블에서 아래의 결과를 출력하는 SELECT 문장을 작성하여라...

TOTAL	1980	1981	1982	1983
14	1	10	2	1

7. EMP 테이블에서 아래의 결과를 출력하는 SELECT 문장을 작성하여라...

JOB	Deptno 10	Deptno 20	Deptno 30	Total
ANALYST		6000		6000
CLERK	1300	1900	950	4150
MANAGER	2450	2975	2850	8275
PRESIDENT	5000			5000
SALESMAN			5600	5600

피벗 테이블

지금까지 GROUP BY를 이용한 부분 집계에 대하여 알아보았다. 예를 들어, 사원 테이블에서 부서, 업무별 급여의 합을 구하기 위해서는 아래와 같은 SQL 문장을 사용했다.

```
SQL> SELECT DEPTNO, JOB, SUM(SAL)
2 FROM EMP
3 GROUP BY DEPTNO, JOB;
```

DEPTNO	JOB	SUM(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	CLERK	1900
20	ANALYST	6000
20	MANAGER	2975
30	CLERK	950
30	MANAGER	2850
30	SALESMAN	5600

	CLERK	MANAGER	PRESIDENT	ANALYST	SALESMAN
10	1300	2450	5000		
20	1900	2975		6000	
30	950	2850			5600

- 1단계 : 원본 테이블에서 피벗 테이블의 가로줄 컬럼을 DECODE를 이용하여 분류한다.

```
SQL> SELECT DEPTNO,
2  DECODE(JOB, 'CLERK', SAL) CLERK,
3  DECODE(JOB, 'MANAGER', SAL) MANAGER,
4  DECODE(JOB, 'PRESIDENT', SAL) PRESIDENT,
5  DECODE(JOB, 'ANALYST', SAL) ANALYST,
6  DECODE(JOB, 'SALESMAN', SAL) SALESMAN
7  FROM EMP;
```

DEPTNO	CLERK	MANAGER	PRESIDENT	ANALYST	SALESMAN
20	800				
30					1600
30					1250
20		2975			
30					1250
30		2850			
10		2450			
20				3000	
10			5000		
30					1500
20	1100				
30	950				
20				3000	
10	1300				

- 2단계 : GROUP BY를 이용하여 부서별로 그룹화한다.

```
SQL> SELECT DEPTNO,  
2  SUM(DECODE(JOB, 'CLERK', SAL)) CLERK,  
3  SUM(DECODE(JOB, 'MANAGER', SAL)) MANAGER,  
4  SUM(DECODE(JOB, 'PRESIDENT', SAL)) PRESIDENT,  
5  SUM(DECODE(JOB, 'ANALYST', SAL)) ANALYST,  
6  SUM(DECODE(JOB, 'SALESMAN', SAL)) SALESMAN  
7  FROM EMP  
8  GROUP BY DEPTNO;
```

DEPTNO	CLERK	MANAGER	PRESIDENT	ANALYST	SALESMAN
10	1300	2450	5000		
20	1900	2975		6000	
30	950	2850			5600


```

SQL> SELECT DEPTNO,
2  SUM(DECODE(JOB, 'CLERK', SAL)) CLERK,
3  SUM(DECODE(JOB, 'MANAGER', SAL)) MANAGER,
4  SUM(DECODE(JOB, 'PRESIDENT', SAL)) PRESIDENT,
5  SUM(DECODE(JOB, 'ANALYST', SAL)) ANALYST,
6  SUM(DECODE(JOB, 'SALESMAN', SAL)) SALESMAN,
7  SUM(SAL) "계"
8  FROM EMP
9  GROUP BY DEPTNO;

```

DEPTNO	CLERK	MANAGER	PRESIDENT	ANALYST	SALESMAN	계
10	1300	2450	5000			8750
20	1900	2975		6000		10875
30	950	2850			5600	9400

마지막으로 맨 마지막줄에 JOB별 합계와 전체 급여 총합을 추가하면 다음과 같다.

```
SQL> SELECT DEPTNO,  
2  SUM(DECODE(JOB, 'CLERK', SAL)) CLERK,  
3  SUM(DECODE(JOB, 'MANAGER', SAL)) MANAGER,  
4  SUM(DECODE(JOB, 'PRESIDENT', SAL)) PRESIDENT,  
5  SUM(DECODE(JOB, 'ANALYST', SAL)) ANALYST,  
6  SUM(DECODE(JOB, 'SALESMAN', SAL)) SALESMAN,  
7  SUM(SAL) "계"  
8  FROM EMP  
9  GROUP BY ROLLUP(DEPTNO);
```

DEPTNO	CLERK	MANAGER	PRESIDENT	ANALYST	SALESMAN	계
10	1300	2450	5000			8750
20	1900	2975		6000		10875
30	950	2850			5600	9400
	4150	8275	5000	6000	5600	29025

1. 사원테이블에서 최대급여, 최소급여, 전체 급여합, 평균을 구하시오
2. 사원테이블에서 부서별 인원수를 구하시오
3. 사원테이블에서 부서별 인원수가 6명 이상인 부서코드를 구하시오
4. 사원테이블에서 다음과 같은 결과가 나오게 하시오

DNAME	CLERK	MANAGER	PRESIDENT	ANALYST	SALESMAN
ACCOUNTING	1300	2450	5000		
RESEARCH	1900	2975		6000	
SALES	950	2850			5600

5. 사원테이블에서 급여가 높은 순서대로 등수를 부여하여 다음과 같은 결과가 나오게 하시오.

힌트 self join, group by, count사용

ENAME	등수
KING	1
FORD	2
SCOTT	2
JONES	4
BLAKE	5
CLARK	6
ALLEN	7
TURNER	8
MILLER	9
MARTIN	10
WARD	10
ADAMS	12
JAMES	13
SMITH	14


결과 집합 내에 집계 값 생성

- ROLLUP 연산자와 함께 GROUP BY 절 사용
 - 그룹 값을 요약하여 출력
- CUBE 연산자와 함께 GROUP BY 절 사용
 - Group By절을 기준으로 그룹의 모든 가능한 조합을 만들고 요약하여 출력

RollUp과 CUBE

- RollUp : 좌우 합계를 내는 형태의 보고서에 사용
- CUBE : 상하 합계를 내는 형태의 보고서에 사용

ROLLUP

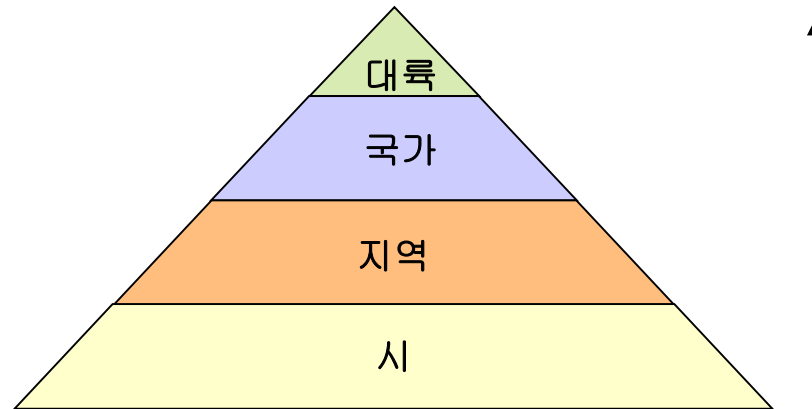


deptno \ job	CLERK	MANAGER	PRESIDENT	ANALYST	SALESMAN	소계
10	1	1	1			3
20	2	1		2		5
30	1	1			4	6
전체합계	4	3	1	2	4	14



CUBE

ROLLUP의 개요



ROLLUP(대륙, 국가, 지역, 시)

\Rightarrow (대륙, 국가, 지역, 시), (대륙, 국가, 지역),
(대륙, 국가), (대륙), ()

※ $\text{ROLLUP}(c_1, c_2, \dots, c_n) \Rightarrow n+1$ 개의 부분합

ROLLUP (1)

- ROLLUP 명령어 GROUP BY 절에 속한 그룹들의 부분합을 계층적으로 구할 수 있게 확장해 준다.

```
SQL> select deptno, sum(sal)
      2  from emp
      3  group by deptno;
```

deptno	SUM(SAL)
10	8750
20	10875
30	9400

ROLLUP (2)

```
SQL> select deptno, sum(sal)
      2  from emp
      3  group by rollup(deptno) ;
```

deptno	SUM(SAL)
10	8750
20	10875
30	9400
	29025

복수개의 컬럼에 대한 ROLLUP

- ROLLUP에서 N개의 컬럼을 지정하였다면 결과화면에서는 N+1개의 그룹을 확인할 수 있다.

```
SQL> select deptno, job, sum(sal)
      2  from emp
      3  group by rollup (deptno, job);
```

DEPTNO	JOB	SUM(SAL)	DEPTNO	JOB	SUM(SAL)
10	CLERK	1300	30	CLERK	950
10	MANAGER	2450	30	MANAGER	2850
10	PRESIDENT	5000	30	SALESMAN	5600
10		8750	30		9400
20	CLERK	1900			29025
20	ANALYST	6000			
20	MANAGER	2975			
20		10875			

ROLLUP을 이용한 또 다른 집합함수

- ROLLUP을 합이 아닌 다른 AGGREGATION 함수에 적용 가능

```
SQL> select deptno, job, avg(sal)
2   from emp
3   group by rollup(deptno, job) ;
```

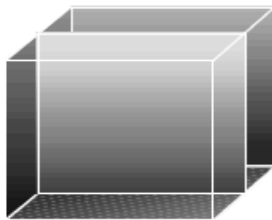
DEPTNO	JOB	AVG (SAL)	DEPTNO	JOB	AVG (SAL)
10	CLERK	1300	30	CLERK	950
10	MANAGER	2450	30	MANAGER	2850
10	PRESIDENT	5000	30	SALESMAN	1400
10		2916.66667	30		1566.66667
20	CLERK	950			2073.21429
20	ANALYST	3000			
20	MANAGER	2975			
20		2175			

CUBE의 개요

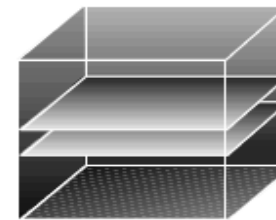
CUBE(A, B, C, D)

\Rightarrow (A, B, C, D), (A, B, C), (A, B, D), (A, C, D), (B, C, D),
(A, B), (A, C), (A, D), (B, C), (B, D), (C, D)
(A), (B), (C), (D), ()

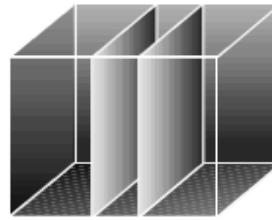
※ CUBE(c1, c2, ...cn) \Rightarrow 2^n 개의 부분합



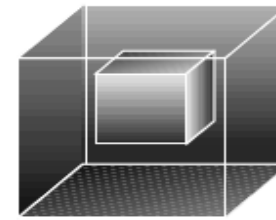
Product Mgr. View



Regional Mgr. View



Financial Mgr. View



Ad Hoc View

결과 집합 내에 집계 값 생성

Aggregate

Sum

Group By (with total)

By Color

RED
WHITE
BLUE

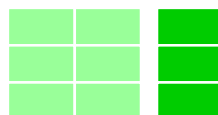


Sum

Cross Tab

Chevy Ford By Color

RED
WHITE
BLUE

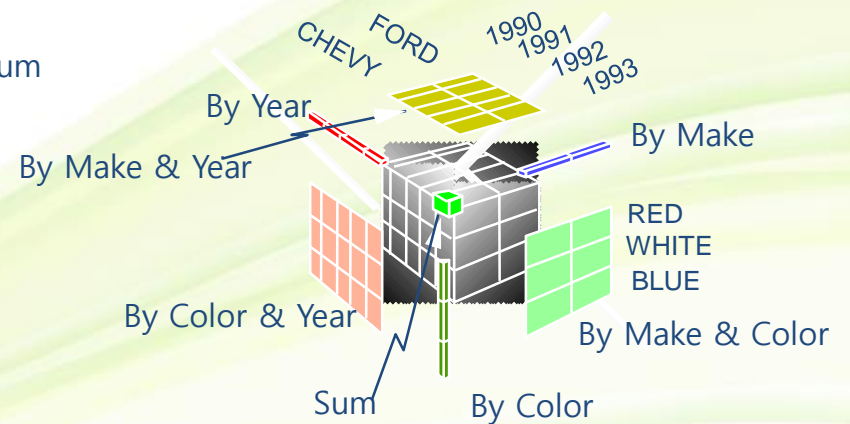


By Make



Sum

The Data Cube and The Sub-Space Aggregates



CUBE

- 선언된 컬럼으로 가능한 모든 조합들의 그룹을 만든다.

```
SQL> select deptno, job, avg(sal)
2   from emp
3   group by cube(deptno, job) ;
```

GROUPING() 함수

- GROUPING() 함수는 한 개의 컬럼을 받아들이고 이 컬럼이 null일 경우 1을, non-null일 경우 0을 리턴.
- GROUPING() 함수는 ROLLUP과 CUBE를 포함한 쿼리에서만 사용.

```
SQL> select grouping(division_id) as grp_id,  
2 division_id, sum(salary)  
3 from employees2  
4 group by rollup(division_id);
```

GRP_ID	DIV	SUM(SALARY)
0	BUS	1610000
0	OPE	1320000
0	SAL	4936000
0	SUP	1015000
1		8881000