

4. 그룹 함수를 이용한 데이터 집계

- ◆ 본 과정을 마치면 다음을 할 수 있어야 합니다.
 - equality join과 nonequality join을 사용하여 하나 이상의 이 용 가능한 그룹 함수를 식별할 수 있어야 합니다.
 - 그룹 함수의 사용을 기술할 수 있어야 합니다.
 - GROUP BY 절을 사용하여 데이터를 그룹화 할 수 있어야 합니다.
 - HAVING 절을 사용하여 그룹 된 행을 포함하거나 제외할 수 있어야 합니다.



그룹 함수란?

- ◆ 그룹 당 하나의 결과가 주어지도록 행의 집합에 대해 연산합니다.

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

EMP 테이블
에서 salary
최대 값

MAX(SAL)
5000

그룹 함수의 유형

함수	의미
AVG	평균 값
SUM	합계
MIN	최소 값
MAX	최대 값
COUNT	행의 수
STDDEV	표준편차
VARIANCE	분산

AVG, SUM, MIN, MAX 함수

- 숫자 데이터에 대해서 AVG, SUM, MIN 그리고 MAX 함수를 사용할 수 있습니다.

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

AVG(SAL)	MAX(SAL)	MIN(SAL)	SUM(SAL)
1400	1600	1250	5600

MIN, MAX 함수 사용

- ◆ 모든 데이터형에 대해서 MIN 과 MAX 함수를 사용할 수 있습니다.

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

MIN(HIRE	MAX(HIRE
80/12/17	87/05/23

COUNT 함수 사용 1

◆ COUNT(*) : 테이블의 행의 수를 리턴합니다.

```
SQL> SELECT COUNT(*)  
2 FROM emp  
3 WHERE deptno = 30;
```

COUNT(*)
6

COUNT 함수 사용 2

- ◆ COUNT(*expr*) : NULL 이 아닌 행의 수를 리턴합니다.

```
SQL> SELECT COUNT(comm)
2 FROM emp
3 WHERE deptno = 30;
```

COUNT(comm)
4

그룹 함수와 NULL값



그룹 함수는 열에 있는 NULL값을 무시합니다.

```
SQL> SELECT AVG(comm)
2 FROM emp;
```

COUNT(comm)
550

그룹 함수와 NVL 함수 사용

- ◆ NVL 함수는 그룹 함수가 NULL 값을 포함하는 것을 가능하게 합니다.

```
SQL> SELECT AVG(NVL(comm,0))  
2 FROM emp;
```

```
AVG(NVL(COMM,0))  
-----  
157.14286
```

데이터 그룹 생성

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

EMP 테이블
에서 부서별
salary 평균
값

DEPTNO	AVG(SAL)
10	2916.66667
20	2175
30	1566.66667

데이터 그룹 생성: GROUP BY 절

- ◆ 테이블의 행을 GROUP BY 절을 사용하여 보다 작은 그룹으로 나눕니다.

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

GROUP BY 절 사용 1



SELECT 절에서 그룹 함수 안에 없는 열들은 GROUP BY 절에 있어야 합니다.

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

DEPTNO	AVG(SAL)
10	2916.66667
20	2175
30	1566.66667

GROUP BY 절 사용 2

- ◆ GROUP BY 절에 정의한 열이 SELECT 절에 꼭 있을 필요는 없습니다.

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

AVG(SAL)
2916.66667
2175
1566.66667

하나 이상의 열로 그룹화

EMP

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

부서 내에서
직업별 급여
(salary) 합

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

다중 열에서 GROUP BY 절 사용

- ◆ GROUP BY 절에 하나 이상의 열을 사용하여 그룹과 서브 그룹에 대한 통계 결과를 리턴할 수 있습니다.

```
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	ANALYST	6000
...		

9 개의 행이 선택되었습니다.

그룹 함수를 잘못 사용한 질의 1



SELECT 절의 그룹 함수가 아닌 모든 열이나 표현식은 GROUP BY 절에 있어야 합니다.

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

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

*

1행에 오류:

ORA-00937: 단일 그룹의 그룹 함수가 아닙니다

GROUP BY 절에 포함된 그룹이 빠져 있습니다.

그룹 함수를 잘못 사용한 질의 2

- ◆ WHERE 절을 사용하여 그룹을 제한할 수 없습니다.
- ◆ 그룹을 제한하기 위해서 HAVING 절을 사용합니다.

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

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

*

3행에 오류:

ORA-00934: 그룹 함수는 허가되지 않습니다

WHERE 절을 사용하여 그룹을 제한할 수 없습니다.

그룹 결과 제외

EMP

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

5000

3000

2850

최대 급여가
2900을 넘는
부서들의 최
대 급여를 계
산함

DEPTNO	SUM(SAL)
10	5000
20	3000

그룹 결과 제외: HAVING 절

- ◆ HAVING 절을 사용하여 그룹을 제한합니다.
 - 행을 그룹화 합니다.
 - 그룹 함수를 적용합니다.
 - HAVING절과 일치하는 그룹을 디스플레이 합니다.

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

HAVING 절 사용 1

- ◆ HAVING 절을 사용하여 그룹을 제한합니다.
 - 행을 그룹화 합니다.
 - 그룹 함수를 적용합니다.
 - HAVING절과 일치하는 그룹을 디스플레이 합니다.

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

DEPTNO	MAX(SAL)
20	3000
10	5000

HAVING 절 사용 2

```
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) ;
```

JOB	PAYROLL
ANALYST	6000
MANAGER	8275

GROUPING SETS 절

- ◆ 오라클 9i에 추가된 문법입니다.
- ◆ GROUP BY절과 UNION ALL연산자의 결합된 형태의 기능입니다.

```
SELECT      column, group_function(column)
FROM        table
WHERE       [condition]
[GROUP BY  [GROUPING SETS] (grouping_expressions)]
[HAVING    group_condition]
[ORDER BY  column];
```

GROUPING SETS 절 사용

```
SQL> SELECT deptno, job, mgr, avg(sal)
2 FROM emp
3 GROUP BY
4 GROUPING SETS ((deptno, job), (job, mgr));
```


DEPTNO	JOB	MGR	AVG(SAL)
10	CLERK		1300
10	MANAGER		2450
10	PRESIDENT		5000
...			
	ANALYST	7566	3000
	CLERK	7698	950
	CLERK	7782	1300

18 개의 행이 선택되었습니다.

ROLLUP과 CUBE

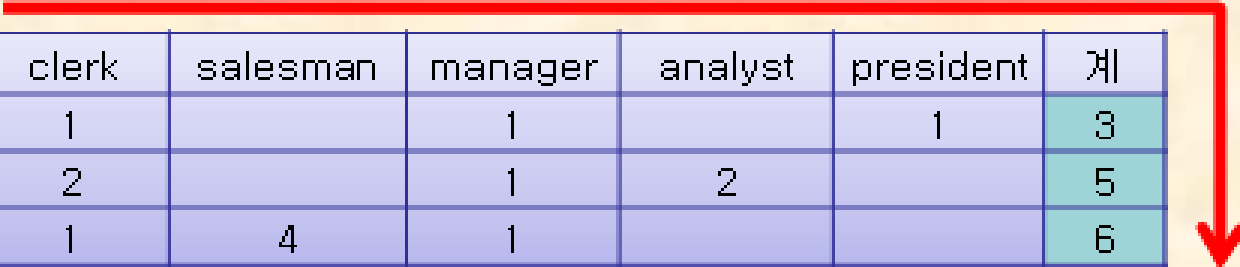
- ◆ Subtotal과 Cross-tab(좌/우, 위/아래 형태의 합계를 내는 형태의 보고서)과 같은 보고서를 만들 때 사용하는 기능입니다.

ROLLUP



deptno	clerk	salesman	manager	analyst	president	계
10	1		1		1	3
20	2		1	2		5
30	1	4	1			6
계						14

CUBE



deptno	clerk	salesman	manager	analyst	president	계
10	1		1		1	3
20	2		1	2		5
30	1	4	1			6
계	4	4	3	2	1	14

ROLLUP 사용

```
SQL> SELECT deptno, GROUPING(deptno),
2          job, GROUPING(job),
3          COUNT(*),
4          AVG(sal)*12
5 FROM emp
6 GROUP BY ROLLUP(deptno, job);
```

DEPTNO	GROUPING(DEPTNO)	JOB	GROUPING(JOB)	COUNT(*)	AVG(SAL)*12
10	0	CLERK	0	1	15600
10	0	MANAGER	0	1	29400
10	0	PRESIDENT	0	1	60000
10	0		1	3	35000
20	0	CLERK	0	2	11400
20	0	ANALYST	0	2	36000
20	0	MANAGER	0	1	35700
20	0		1	5	26100
30	0	CLERK	0	1	11400
30	0	MANAGER	0	1	34200
30	0	SALESMAN	0	4	16800
30	0		1	6	18800
	1		1	14	24878.5714

13 개의 행이 선택되었습니다.

CUBE 사용

```
SQL> SELECT deptno, GROUPING(deptno),
2         job, GROUPING(job),
3         COUNT(*),
4         AVG(sal)*12
5 FROM emp
6 GROUP BY CUBE(deptno, job);
```

DEPTNO	GROUPING(DEPTNO)	JOB	GROUPING(JOB)	COUNT(*)	AVG(SAL)*12
	1		1	14	24878.5714
	1	CLERK	0	4	12450
	1	ANALYST	0	2	36000
	1	MANAGER	0	3	33100
	1	SALESMAN	0	4	16800
	1	PRESIDENT	0	1	60000
10	0		1	3	35000
10	0	CLERK	0	1	15600
10	0	MANAGER	0	1	29400
10	0	PRESIDENT	0	1	60000
20	0		1	5	26100
DEPTNO	GROUPING(DEPTNO)	JOB	GROUPING(JOB)	COUNT(*)	AVG(SAL)*12
20	0	CLERK	0	2	11400
20	0	ANALYST	0	2	36000
20	0	MANAGER	0	1	35700
30	0		1	6	18800
30	0	CLERK	0	1	11400
30	0	MANAGER	0	1	34200
30	0	SALESMAN	0	4	16800

18 개의 행이 선택되었습니다.

Quiz 1

- SELECT 문의 결과에서 group을 제한하기 위해 사용하는 절은?
 - A. WHERE
 - B. HAVING
 - C. ORDER BY
 - D. GROUP BY

Quiz 2

- 다음 SQL 명령문에서 ERROR가 발생하는 행은?
 - A. SELECT ename, sum(sal) "Salary"
 - B. FROM emp
 - C. WHERE sal > 1000
 - D. ORDER BY 2;

Quiz 3

- 다음 SQL문장에서 틀린 곳을 지적하십시오.
 - A. SELECT deptno, AVG(sal)
 - B. FROM emp
 - C. WHERE AVG(sal) > 1000
 - D. GROUP BY deptno;

Quiz 4

- 다음 중 Group function 수행 시 NULL 값을 포함하여 연산하는 경우는?
- A. `SELECT MIN(ename), MAX(ename)`
`FROM emp;`
 - B. `SELECT AVG(sal)`
`FROM emp WHERE UPPER(job) LIKE 'VP%';`
 - C. `SELECT COUNT(*)`
`FROM emp WHERE deptno = 20;`
 - D. `SELECT SUM(sal)`
`FROM emp WHERE DEPTNO = 30;`

Quiz 5

- 다음 중 null 값을 포함하는 행을 연산에 포함시키는 함수는?
- A. COUNT(*)
 - B. AVG
 - C. MAX
 - D. SUM

Quiz 6

- 모든 DATATYPE에 가능한 GROUP 함수들은?
 - A. AVG
 - B. MIN
 - C. MAX
 - D. COUNT
 - E. STDDEV
 - F. VARIANCE