



SQL 활용

집합 연산자와 집단 연산자



한국기술교육대학교
온라인평생교육원

학습내용

- 집합 연산자
- 집단 연산자

학습목표

- 집합 연산자를 이용하여 질의문을 작성할 수 있다.
- 집단 함수와 집단 연산자를 이용하여 다양한 통계정보를 추출할 수 있다.

● 집합 연산자

1. 집합 연산자

◆ 집합 연산자

- 테이블을 구성하는 튜플 집합에 대한 테이블의 부분 집합을 결과로 반환하는 연산자
- UNION : 합집합
- INTERSECT : 교집합
- EXCEPT : 차집합(Oracle에선 MINUS로 사용)

◆ UNION의 예

Q 부서 번호 10인 직원들과 직급이 staff인 직원들 검색

```
USE MagicCorp
GO

(select *
FROM EMPLOYEE
WHERE DNO = 10)
UNION
(select *
FROM EMPLOYEE
WHERE JOB = 'staff')
```

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	101	e1	staff	113	2007-03-01 00:00:00.000	300	NULL	20
2	107	e7	chief	108	2004-01-08 00:00:00.000	520	NULL	10
3	110	e10	section	103	2005-04-07 00:00:00.000	500	NULL	10
4	111	e11	staff	107	2007-03-01 00:00:00.000	280	NULL	30
5	112	e12	staff	106	2007-08-09 00:00:00.000	300	NULL	20
6	114	e14	staff	106	2007-11-09 00:00:00.000	250	NULL	10

● 집합 연산자

1. 집합 연산자

◆ INTERSECT의 예

Q 부서 번호 10이고 직급이 staff인 사원들 검색

```
USE MagicCorp
GO

(select *
FROM EMPLOYEE
WHERE DNO = 10)
INTERSECT
(select *
FROM EMPLOYEE
WHERE JOB = 'staff')
```

Results Messages

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	114	e14	staff	106	2007-11-09 00:00:00.000	250	NULL	10

◆ EXCEPT의 예

Q 부서 번호 10인 사원들 중 직급이 staff인 사원들 검색

```
USE MagicCorp
GO

(select *
FROM EMPLOYEE
WHERE DNO = 10)
EXCEPT
(select *
FROM EMPLOYEE
WHERE JOB = 'staff')
```

100 % <

결과 메시지

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	107	e7	chief	108	2004-01-08 00:00:00.000	520	NULL	10
2	110	e10	section	103	2005-04-07 00:00:00.000	500	NULL	10

● 집합 연산자

2. UNION과 UNION ALL

집합 연산자를 대상 테이블을 집합으로 봄

- 따라서 결과도 집합임 ⇨ 중복을 허용하지 않음

필요에 따라서 중복된 결과도 보고 싶은 경우



UNION ALL을 사용

◆ UNION ALL의 사용

Q 부서 번호 10인 직원들과 직급이 staff인 직원들 검색(중복 허용)

```
USE MagicCorp
GO

(select *
FROM EMPLOYEE
WHERE DNO = 10)
UNION ALL
(select *
FROM EMPLOYEE
WHERE JOB = 'staff')
```

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	107	e7	chief	108	2004-01-08 00:00:00.000	520	NULL	10
2	110	e10	section	103	2005-04-07 00:00:00.000	500	NULL	10
3	114	e14	staff	106	2007-11-09 00:00:00.000	250	NULL	10
4	101	e1	staff	113	2007-03-01 00:00:00.000	300	NULL	20
5	111	e11	staff	107	2007-03-01 00:00:00.000	280	NULL	30
6	112	e12	staff	106	2007-08-09 00:00:00.000	300	NULL	20
7	114	e14	staff	106	2007-11-09 00:00:00.000	250	NULL	10

● 집합 연산자

3. 외부 합집합

◆ 합병 호환성

$U, \cap, -$ 연산의 피연산자(릴레이션)들이 지켜야 할 제약 조건

- 차수(Degree : 속성의 수)가 같아야 함
- 대응되는 속성 쌍 별로 타입(또는 도메인)이 같아야 함
- 대응되는 속성 쌍 별로 의미(Semantic)가 같아야 함

◆ 합병 호환성의 불일치

두 질의 결과가 합병 호환성을 만족하지 않음

합집합하고 싶은 경우



외부 합집합($U+$)

◆ SQL에서의 외부 합집합

- 두 질의 결과의 속성 수와 타입이 일치되게 만듦
 - NULL은 모든 속성에서 사용할 수 있는 속성값임
- ⇒ 이를 이용하여 강제로 동일한 구조가 되게 함

● 집합 연산자

3. 외부 합집합

◆ SQL에서의 외부 합집합

- 두 질의 결과의 속성 수와 타입이 일치되게 만듦
- NULL은 모든 속성에서 사용할 수 있는 속성값임

⇒ 이를 이용하여 강제로 동일한 구조가 되게 함

● 외부 합집합의 예

- 두 질의 결과에서 공통인 속성은 dno밖에 없음
- 다른 것들은 NULL로 출력하도록 함

⇒ 속성의 수 : 각각 4개, 타입도 일치함

⇒ 합집합 할 수 있음

```
(select eno, ename, dno , NULL from EMPLOYEE)
UNION
(select NULL, NULL, dno, dname from DEPARTMENT)
```

Results									
	eno	ename	dno	(No column name)		eno	ename	dno	(No column name)
1	101	e1	20	NULL	10	110	e10	10	NULL
2	102	e2	30	NULL	11	111	e11	30	NULL
3	103	e3	30	NULL	12	112	e12	20	NULL
4	104	e4	20	NULL	13	113	e13	20	NULL
5	105	e5	30	NULL	14	114	e14	10	NULL
6	106	e6	30	NULL	15	NULL	NULL	10	Accounting
7	107	e7	10	NULL	16	NULL	NULL	20	Human
8	108	e8	30	NULL	17	NULL	NULL	30	Sales
9	109	e9	20	NULL	18	NULL	NULL	40	Computing


● 집단 연산자

1. 집단 함수

◆ 집단 함수란?

- 테이블의 전체 행을 하나 이상의 컬럼을 기준으로 그룹화하여 해당 그룹 별 통계 값을 출력하는 함수

Q 전체 직원의 평균 봉급을 구하기



The screenshot shows a SQL query window with the following text:

```
USE MagicCorp
GO

SELECT AVG(SALARY)
FROM EMPLOYEE
```

Below the query window, the 'Results' tab is active, displaying a single row of data:

	(No column name)
1	463

◆ 집단 함수의 종류

- SUM : 그룹의 합계
- AVG : 그룹의 평균
- COUNT : 그룹의 개수
- MAX : 그룹의 최대값
- MIN : 그룹의 최소값
- STDEV : 그룹의 표준편차
- VAR : 그룹의 분산

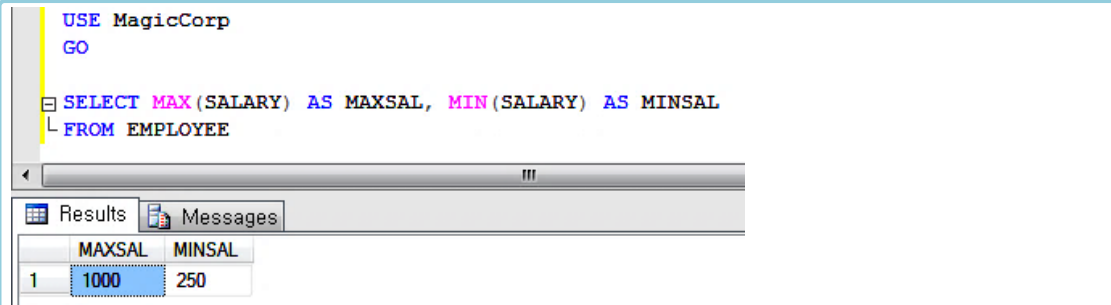
일반적으로 집단 함수는 NULL값을 제외한 속성값들의 통계 값을 반환함

● 집단 연산자

1. 집단 함수

◆ 집단 함수의 예

Q 사원의 최대 봉급, 최소 봉급 구하기



The screenshot shows a SQL query window with the following text:

```
USE MagicCorp
GO

SELECT MAX(SALARY) AS MAXSAL, MIN(SALARY) AS MINSAL
FROM EMPLOYEE
```

Below the query window, there is a 'Results' tab showing the output of the query. The results are displayed in a table with two columns: MAXSAL and MINSAL. The first row shows the values 1000 and 250.

	MAXSAL	MINSAL
1	1000	250

● 집단 연산자

1. 집단 함수

◆ 분산(VAR)

- 각 값이 평균과 얼마나 떨어져 있는지에 대한 통계 값
- 각 값과 평균의 차에 대한 차(즉, 편차)의 제곱의 평균
 - 편차는 음수이거나 양수 일수 있으므로 편차를 제곱하여 양수로 나타내고 이들의 합에 대한 평균을 구함

$$(\sum_{i=1,n} (x_i - M)^2) / N$$

- x_i : 각 값
- M : 평균
- N : 값의 개수

◆ 표준 편차(STDEV)

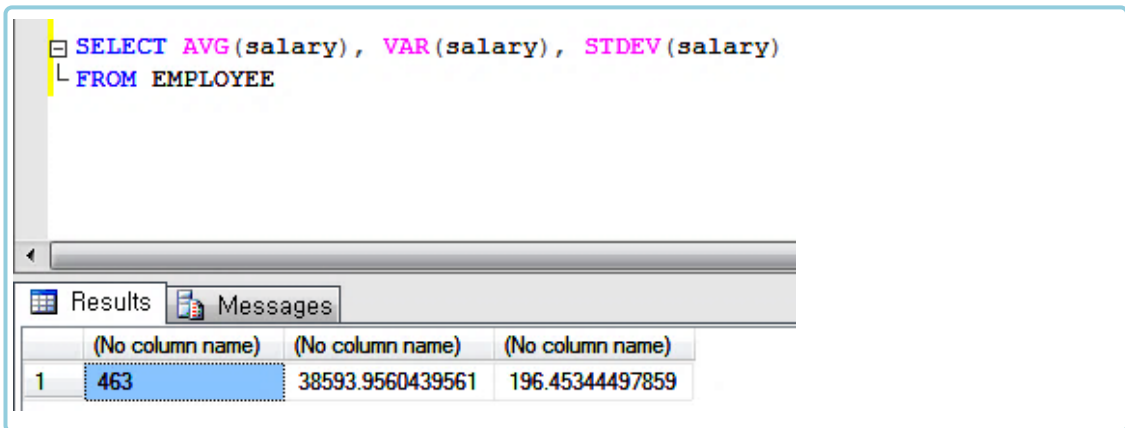
- 분산의 경우 편차에 대한 제곱으로 나타내므로 평균과의 단위가 맞지 않음
- 단위를 맞추기 위하여 분산의 제곱근을 표준편차로 사용함

$$STDEV = VAR^{1/2}$$

◆ 분산(VAR)과 표준 편차(STDEV)

Q 사원들의 급여 평균, 분산, 표준편차 구하기

- NULL값을 제외한 모든 행의 평균, 분산, 표준편차가 구해짐



```
SELECT AVG(salary), VAR(salary), STDEV(salary)
FROM EMPLOYEE
```

	(No column name)	(No column name)	(No column name)
1	463	38593.9560439561	196.45344497859

● 집단 연산자

1. 집단 함수

◆ COUNT(*)와 COUNT(속성명)

● COUNT(*)

- 테이블에서 조건을 만족하는 행의 개수를 반환하는 함수
- *는 모든 속성들이란 의미임

● COUNT(속성명)

- 속성값이 NULL이 아닌 속성값의 개수

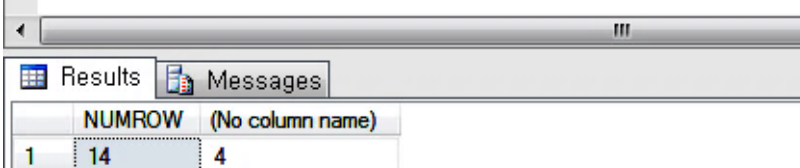
● COUNT(DISTINCT 속성명)

- 속성값이 NULL이 아니며 중복되지 않는 속성값들의 개수
- SUM, AVG도 DISTINCT를 쓸 수 있음
- MIN, MAX에서는 DISTINCT가 의미가 없음

Q 사원 테이블의 튜플수와 COMMISSION의 개수 구하기

```
USE MagicCorp
GO

SELECT COUNT(*) AS NUMROW, COUNT(COMMISSION)
FROM EMPLOYEE
```




	NUMROW	(No column name)
1	14	4

Q 사원의 직급(JOB)의 수와 중복되지 않는 직급(JOB)의 수 구하기

```
USE MagicCorp
GO

SELECT COUNT(JOB) AS NUMJOB, COUNT(DISTINCT JOB) AS NUMDISTINCTJOB
FROM EMPLOYEE
```



	NUMJOB	NUMDISTINCTJOB
1	14	6

● 집단 연산자

2. GROUP BY와 HAVING

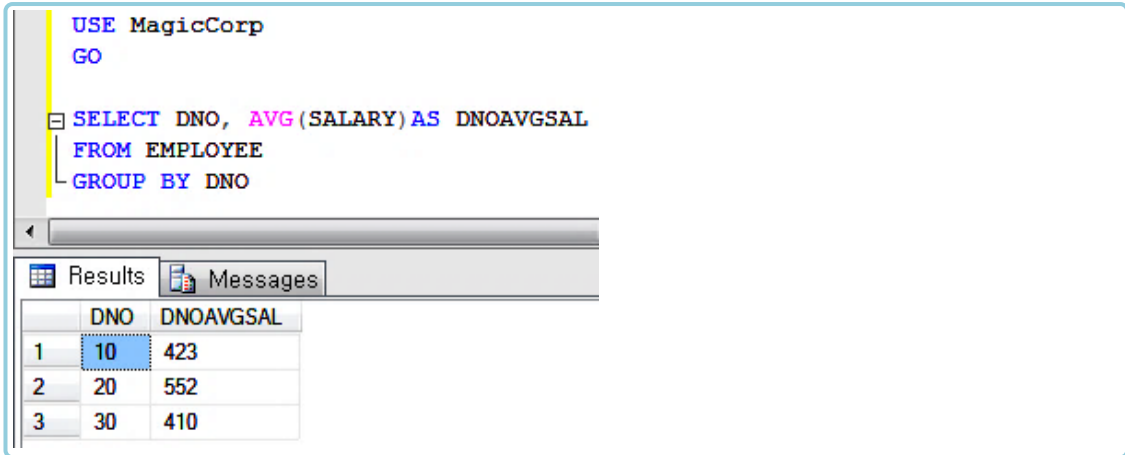
◆ GROUP BY

- 특정 속성을 기준으로 테이블 전체를 그룹으로 나누기 위한 절

예) 부서별 사원의 평균 봉급

```
SELECT 컬럼_리스트  
FROM 테이블명  
WHERE 조건  
GROUP BY 컬럼_리스트
```

Q 부서별 사원들의 평균 봉급과 부서번호 검색하기



The screenshot shows a SQL query window with the following code:

```
USE MagicCorp  
GO  
  
SELECT DNO, AVG(SALARY) AS DNOAVGSAL  
FROM EMPLOYEE  
GROUP BY DNO
```

Below the query window, the 'Results' tab is active, displaying the following data:

	DNO	DNOAVGSAL
1	10	423
2	20	552
3	30	410

● 집단 연산자

2. GROUP BY와 HAVING

◆ GROUP BY 사용 시 주의점

- ① SELECT 절에는 집단 연산자나 GROUP BY에 사용한 속성명만을 사용할 수 있음
- ② 공통되는 속성값으로 그룹핑을 했으므로, 각 그룹에서 개별 튜플을 접근할 수 없음

Q 부서별 직원들의 평균 봉급과 부서번호, 직원이름 검색하기(오류)

```
USE MagicCorp
GO

SELECT DNO, AVG(SALARY) AS DNOAVGSAL, ENAME
FROM EMPLOYEE
GROUP BY DNO
```

Messages

Msg 8120, Level 16, State 1, Line 2
Column 'EMPLOYEE.ENAME' is invalid in the select list because it is not contained in eit

◆ HAVING

- 각 그룹에 대한 제약 조건을 기술할 때 사용함
- HAVING 절은 GROUP BY 절의 종속절임
 - GROUP BY없이 HAVING은 나타낼 수 없음
- WHERE 절은 테이블 전체에 대한 제약 조건을 나타냄

Q 부서의 최대 봉급이 500초과인 부서에 대해서만 부서별 직원들의 평균 봉급과 부서 번호 출력하기

```
USE MagicCorp
GO

SELECT DNO, AVG(SALARY) AS DNOAVGSAL
FROM EMPLOYEE
GROUP BY DNO
HAVING MAX(SALARY) > 500
```

Results

	DNO	DNOAVGSAL
1	10	423
2	20	552

● 집단 연산자

3. ROLLUP과 CUBE

◆ 다중 속성 GROUP BY

- 하나 이상의 속성들을 이용하여 그룹을 나누고, 그룹별로 다시 서브 그룹을 나누고자 할 때 사용함

GROUP BY 컬럼1, 컬럼2, ..., 컬럼ⁿ

Q 부서별 직급별 직원들의 봉급 합 구하기

```
USE MagicCorp
GO

select dno, job, sum(salary) as [AVGSAL]
from employee
group by dno, job
```

	dno	job	AVGSAL
1	20	ceo	1000
2	10	chief	520
3	20	chief	1160
4	30	chief	480
5	30	deputy	250
6	10	section	500
7	30	section	950
8	30	senior	500
9	10	staff	250
10	20	staff	600
11	30	staff	280

● 집단 연산자

3. ROLLUP과 CUBE

◆ ROLLUP 연산자

- GROUP BY절의 그룹 조건에 따라서 그룹핑 하고 각 그룹에 대한 부분합을 구하는 연산자
- GROUP BY절에 n개의 속성 명이 있으면, n+1개의 그룹핑 조합이 나옴
- GROUP BY c1, c2, c3일 때
 - 각 그룹 c1, c2, c3 별 합
 - 각 그룹 c1, c2 별 합
 - 그룹 c1 별 합
 - 전체 합

Q ROLLUP 연산자를 이용하여 부서별 직급별 직원들의 봉급 합 구하기

```
USE MagicCorp
GO

select dno, job, sum(salary) as [totalSAL]
from employee
group by dno, job with rollup
```

	dno	job	totalSAL
1	10	chief	520
2	10	section	500
3	10	staff	250
4	10	NULL	1270
5	20	ceo	1000
6	20	chief	1160
7	20	staff	600
8	20	NULL	2760
9	30	chief	480
10	30	deputy	250
11	30	section	950
12	30	senior	500
13	30	staff	280
14	30	NULL	2460
15	NULL	NULL	6490

● 집단 연산자

3. ROLLUP과 CUBE

◆ CUBE 연산자

- GROUP BY 절의 그룹 조건에 따라서 그룹핑하고 각 그룹의 조합에 따른 부분합을 구하는 연산자
- GROUP BY 절에 n 개의 속성명이 있으면 2^n 개의 그룹핑 조합이 나옴
- GROUP BY $c1, c2, c3$ 일 때
 - 각 그룹 $c1, c2, c3$ 별 합
 - 각 그룹 $c1, c2$ 별 합, $c1, c3$ 별 합, $c2, c3$ 별 합
 - 그룹 $c1$ 별 합, $c2$ 별 합, $c3$ 별 합
 - 전체 합

Q CUBE 연산자를 이용하여 부서별 직급별 직원들의 봉급 합 구하기

```
USE MagicCorp
GO

select dno, job, sum(salary) as [totalSAL]
from employee
group by dno, job with cube
```

	dno	job	totalSAL		dno	job	totalSAL
1	20	ceo	1000	12	30	senior	500
2	NULL	ceo	1000	13	NULL	senior	500
3	10	chief	520	14	10	staff	250
4	20	chief	1160	15	20	staff	600
5	30	chief	480	16	30	staff	280
6	NULL	chief	2160	17	NULL	staff	1130
7	30	deputy	250	18	NULL	NULL	6490
8	NULL	deputy	250	19	10	NULL	1270
9	10	section	500	20	20	NULL	2760
10	30	section	950	21	30	NULL	2460
11	NULL	section	1450				

- 집단 연산자

3. ROLLUP과 CUBE

- ◆ GROUPING SETS 함수

여러 개의 GROUP조건을 표시하고 싶은 경우

예 부서 별 급여 합과 직급별 급여 합을 한번에 보고 싶음

부서, 직급별 합을 보고 싶지 않은 경우



GROUPING SETS 함수를 이용함

```
select dno, job, sum(salary) as [totalSAL]
from employee
group by GROUPING SETS( (dno), (job) )
```

Results

Messages

	dno	job	totalSAL
1	NULL	ceo	1000
2	NULL	chief	2160
3	NULL	deputy	250
4	NULL	section	1450
5	NULL	senior	500
6	NULL	staff	1130
7	10	NULL	1270
8	20	NULL	2760
9	30	NULL	2460

← GROUP BY job의 결과

← GROUP BY dno의 결과

핵심요약

1. 집합 연산자

■ 집합 연산자

- 테이블을 구성하는 튜플 집합에 대한 테이블의 부분 집합을 결과로 반환하는 연산자
 - UNION : 합집합
 - INTERSECT : 교집합
 - EXCEPT : 차집합(Oracle에선 MINUS로 사용)

■ UNION과 UNION ALL

- 집합 연산자를 대상 테이블을 집합으로 봄
- 따라서 결과도 집합임
 - 중복을 허용하지 않음
- 필요에 따라서 중복된 결과도 보고 싶은 경우
 - UNION ALL을 사용함

■ 외부 합집합

- 합병 호환성
 - \cup , \cap , - 연산의 피연산자(릴레이션)들이 지켜야 할 제약 조건
- 합병 호환성의 불일치
 - 두 질의 결과는 합병 호환성을 만족하지 않음
 - 합병 호환성이 만족되지 않는 두 테이블의 합집합 구하기 : 외부 합집합(U^+)
- SQL에서의 외부 합집합
 - 두 질의 결과의 속성 수와 타입이 일치되게 만듦
 - NULL은 모든 속성에서 사용할 수 있는 속성값임
 - 이를 이용하여 강제로 동일한 구조가 되게 함

핵심요약

2. 집단 연산자

■ 집단 함수

- 테이블의 전체 행을 하나 이상의 컬럼을 기준으로 그룹화 하여 그 그룹 별 통계값을 출력하는 함수

- SUM : 그룹의 합계
- AVG : 그룹의 평균
- COUNT : 그룹의 개수
- MAX : 그룹의 최대값
- MIN : 그룹의 최소값
- STDEV : 그룹의 표준편차
- VAR : 그룹의 분산

■ 분산(VAR)

- 각 값이 평균과 얼마나 떨어져 있는지에 대한 통계값
- 각 값과 평균의 차에 대한 차(즉, 편차)의 제곱의 평균

$$(\sum_{i=1,n} (x_i - M)^2) / N$$

■ 표준 편차(STDEV)

- 분산의 경우 편차에 대한 제곱으로 나타냄으로 평균과의 단위가 맞지 않음
- 단위를 맞추기 위하여 분산의 제곱근을 표준편차로 사용함

$$STDEV = VAR^{1/2}$$

핵심요약

2. 집단 연산자

■ 집단 함수

■ COUNT(*)

- 테이블에서 조건을 만족하는 행의 개수를 반환하는 함수

■ COUNT(속성명)

- 속성값이 NULL아닌 속성값의 개수

■ COUNT(DISTINCT 속성명)

- 속성값이 NULL이 아니며 중복되지 않는 속성값들의 개수

■ GROUP BY와 HAVING

■ GROUP BY

- 특정 속성을 기준으로 테이블 전체를 그룹으로 나누기 위한 절

```
SELECT 컬럼_리스트  
FROM 테이블명  
WHERE 조건  
GROUP BY 컬럼_리스트
```

- SELECT 절에는 집단 연산자나 GROUP BY에 사용한 속성명 만을 사용할 수 있음

- 공통되는 속성값으로 그룹핑을 했으므로, 각 그룹에서 개별 튜플을 접근할 수 없음

■ HAVING

- 각 그룹에 대한 제약 조건을 기술할 때 사용함

- HAVING 절은 GROUP BY 절의 종속절임

- WHERE 절은 테이블 전체에 대한 제약 조건을 나타냄

핵심요약

2. 집단 연산자

■ ROLLUP과 CUBE

■ 다중 속성 GROUP BY

- 하나 이상의 속성들을 이용하여 그룹을 나누고, 그룹별로 다시 서브 그룹을 나누고자 할 때

GROUP BY 컬럼1, 컬럼2, ..., 컬럼ⁿ

■ ROLLUP 연산자

- GROUP BY 절의 그룹 조건에 따라서 그룹핑 하고 각 그룹에 대한 부분합을 구하는 연산자
- GROUB BY절에 n개의 속성 명이 있으면, n+1개의 그룹핑 조합이 나옴

■ CUBE 연산자

- GROUP BY 절의 그룹 조건에 따라서 그룹핑하고 각 그룹의 조합에 따른 부분합을 구하는 연산자
- GROUB BY 절에 n 개의 속성명이 있으면 2n개의 그룹핑 조합이 나옴

■ GROUPING SETS 함수

- 경우에 따라서 여러 개의 GROUP조건을 표시하고 싶은 경우, 부서,직급별 합을 보고 싶지 않은 경우 GROUPING SETS 함수를 이용함