Introduction to Oracle9i
: SQL

Chapter12.
향상된 GROUP BY 절

▶ 그룹핑의 향상된 기능을 학습한다.
▶ ROLLUP, CUBE, GROUPING, GROUPING SETS의 활용용도를 이해한다.

1. 그룹함수

```
SELECT     [컬럼명,] 그룹함수(컬럼명)...
FROM       테이블명
[WHERE     조건]
[GROUP BY  컬럼명]
[ORDER BY  컬럼명] ;
```

```
SELECT     AVG(salary), COUNT(commission_pct), MAX(hire_date)
FROM       employees
WHERE      job_id LIKE '%MAN%' ;
```

2. GROUP BY 절

```
SELECT     [컬럼명,] 그룹함수(컬럼명)...
FROM       테이블명
[WHERE     조건]
[GROUP BY  컬럼명]
[ORDER BY  컬럼명] ;
```

```
SELECT      department_id, job_id, SUM(salary)
FROM        employees
GROUP BY    department_id, job_id ;
```

3. HAVING 절

```
SELECT     [컬럼명,] 그룹함수(컬럼명)...
FROM       테이블명
[WHERE     조건]
[GROUP BY  컬럼명]
[HAVING    표현식]
[ORDER BY  컬럼명] ;
```

```
SELECT     department_id, SUM(salary)
FROM       employees
GROUP BY   department_id
HAVING     SUM(salary) > 15000 ;
```

4. 향상된 GROUP BY절 활용

(1) ROLLUP

- GROUP BY 절의 향상된 기능
- 중간합계와 같은 누적 총계값을 산출

```
SELECT      [컬럼명,] 그룹함수(컬럼명)...
FROM        테이블명
[WHERE      조건]
[GROUP BY  ROLLUP  컬럼명]
[HAVING  표현식]
[ORDER BY  컬럼명]
```

4. 향상된 GROUP BY절 활용

(1) ROLLUP

```
SELECT      department_id, job_id, SUM(salary)
FROM        employees
WHERE       departmrnt_id < 90
GROUP BY  department_id, job_id ;
```

```
SELECT      department_id, job_id, SUM(salary)
FROM        employees
WHERE       departmrnt_id < 90
GROUP BY  ROLLUP (department_id, job_id) ;
```

4. 향상된 GROUP BY절 활용

(2) CUBE

- GROUP BY 절의 향상된 기능
- 교차 누적 총계값을 산출

```
SELECT      [컬럼명,] 그룹함수(컬럼명)…
FROM        테이블명
[WHERE      조건]
[GROUP BY  CUBE  컬럼명]
[HAVING   표현식]
[ORDER BY  컬럼명] ;
```

4. 향상된 GROUP BY절 활용

(2) CUBE

```
SELECT     department_id, job_id, SUM(salary)
FROM       employees
WHERE      departmrnt_id < 90
GROUP BY   department_id, job_id ;
```

```
SELECT     department_id, job_id, SUM(salary)
FROM       employees
WHERE      departmrnt_id < 90
GROUP BY   CUBE (department_id, job_id) ;
```

4. 향상된 GROUP BY절 활용

(3) GROUPING 함수

- ROLLUP 또는 CUBE 연산과 함께 사용
- 0 또는 1 값을 반환
  - 0 : 총계값 계산에 사용
  - 1 : 총계값 계산에 비사용

```
SELECT     [컬럼명,] 그룹함수(컬럼명)..., GROUPING(표현식)
FROM       테이블명
[WHERE     조건]
[GROUP BY  [ROLLUP] [CUBE] 컬럼명]
[HAVING   표현식]
[ORDER BY  컬럼명] ;
```

4. 향상된 GROUP BY절 활용

(3) GROUPING 함수

```
SELECT      department_id DEPT, job_id JOB, SUM(salary),
            GROUPING(department_id) GRP_DEPT,
            GROUPING(job_id) GRP_JOB
FROM        employees
WHERE       departmrnt_id < 50
GROUP BY    ROLLUP (department_id, job_id) ;
```

DEPTID	JOB	SUM(SALARY)	GRP_DEPT	GRP_JOB
10	AD_ASST	4400	0	0
10		4400	0	1
20	MK_MAN	13000	0	0
20	MK_REP	6000	0	0
20		19000	0	1
		23400	1	1

4. 향상된 GROUP BY절 활용

(4) GROUPING SETS

- GROUP BY의 우선순위를 다양하게 조정
- 각각의 GROUPING 결과를 UNION ALL 한 결과

```
SELECT      department_id, job_id, manager_id, AVG(salary)
FROM        employees
GROUP BY    GROUPING SETS
( (department_id, job_id), (job_id, manager_id) ) ;
```

=

```
SELECT      department_id, job_id, NULL as manager_id, AVG(salary)
FROM        employees
GROUP BY    department_id, job_id
UNION ALL
SELECT      NULL, job_id, manager_id, AVG(salary)
FROM        employees
GROUP BY     job_id, manager_id ;
```

4. 향상된 GROUP BY절 활용

(5) 복합 컬럼의 사용

GROUPING SETS	동등한 GROUP BY문

GROUP BY GROUPING SETS (a, b, c)	GROUP BY a UNION ALL
	GROUP BY b UNION ALL
	GROUP BY c

GROUP BY GROUPING SETS (a, b, (b, c))	GROUP BY a UNION ALL
	GROUP BY b UNION ALL
	GROUP BY b, c

GROUP BY GROUPING SETS ((a, b, c))	GROUP BY a, b, c

GROUP BY GROUPING SETS (a, (b), ())	GROUP BY a UNION ALL
	GROUP BY b UNION ALL
	GROUP BY ()

GROUP BY GROUPING SETS
(a, ROLLUP(b, c))
	GROUP BY a UNION ALL
	GROUP BY ROLLUP(b, c)

4. 향상된 GROUP BY절 활용

(5) 복합 컬럼의 사용

```
SELECT      department_id, job_id, manager_id, SUM (salary)
FROM        employees
GROUP BY   ROLLUP  (department_id, (job_id, manager_id) ) ;
```

=

```
SELECT      department_id, job_id, manager_id, SUM(salary)
FROM        employees
GROUP BY   department_id, job_id, manager_id
UNION  ALL
SELECT      department_id, TO_CHAR(NULL), TO_NUMBER(NULL), SUM(salary)
FROM        employees
GROUP BY   department_id
UNION  ALL
SELECT TO_NUMBER(NULL),TO_CHAR(NULL),TO_NUMBER(NULL), SUM(salary)
FROM        employees
GROUP BY   ( ) ;
```

4. 향상된 GROUP BY절 활용

(6) 연쇄 GROUPING

- 유용한 GROUPING의 결합을 수행하는 쉬운 방법을 제공
- 연쇄 GROUPING SET을 지정하기 위해 다중 GROUPING SET, ROLLUP, CUBE 연산을 콤마 기호와 함께 사용
- 결과물은 각각의 GROUPING SET으로부터 추출한 GROUPING의 교차 산출물

4. 향상된 GROUP BY절 활용

(6) 연쇄 GROUPING

```
SELECT      department_id, job_id, manager_id, SUM (salary)
FROM        employees
GROUP BY    department_id, ROLLUP (job_id), CUBE(manager_id) ;
```

```
Total salary for every department_id, job_id, manager_id
Total salary for every department_id, manager_id
Total salary for every department_id, job_id
Total salary for every department_id
```