

실습 8. (집계함수+문자열함수) 활용 / 그룹핑컬럼 가공

8-1. 부서별 급여를 가장 많이 받는 사원의 이름(first_name+last_name)과 급여(salary)

```
select a.dept_name
      , substr(max(lpad(d.salary, 6, '0')||c.first_name||' '||c.last_name), 7) emp_name
      , max(d.salary) salary
from departments a, dept_emp b, employees c, salaries d
where a.dept_no = b.dept_no and b.to_date = '99991231'
      and b.emp_no = c.emp_no
      and c.emp_no = d.emp_no and d.to_date = '99991231'
group by a.dept_name ;
```

출력 결과

DEPT_NAME	EMP_NAME	SALARY
1 Research	Ramachenga Soicher	130211
2 Development	Khosrow Sgarro	144434
3 Quality Management	Shin Luck	132103
4 Human Resources	Yinlin Flowers	141953
5 Customer Service	Vidya Hanabata	144866
6 Production	Youjian Cronau	138273
7 Finance	Lunjin Swick	142395
8 Sales	Tokuyasu Pesch	158220
9 Marketing	Akemi Warwick	145128

실습 8. (집계함수+문자열함수) 활용 / 그룹핑컬럼 가공

8-2. 현직 사원에 대한 입사연도별 급여 평균

```
select /*+ opt_param('_GBY_HASH_AGGREGATION_ENABLED' 'false') */ substr(to_char(a.hire_date,
'yyyymmdd'), 1, 4) 입사연도
      , round(avg(b.salary)) 급여평균
from employees a, salaries b
where a.emp_no = b.emp_no
      and b.to_date = '99991231'
group by substr(to_char(a.hire_date, 'yyyymmdd'), 1, 4) ;
```

입사연도 순으로 정렬된 결과를 얻기 위해서
Sort group by를 활용하기 위한 hint

출력 결과

입사연도	급여평균
1 1985	78870
2 1986	77411
3 1987	75928
4 1988	74202
5 1989	73053
6 1990	71484
7 1991	69813
8 1992	68286
9 1993	67091
10 1994	65333
11 1995	63705
12 1996	62425
13 1997	60795
14 1998	59673
15 1999	58199
16 2000	58192

실습 9. UNION ALL을 활용한 실행계획 분리

9-1. 사원 이름으로 사원정보(성명, 연령, 입사일자, 소속부서명, 직급명, 급여) 검색 (first_name으로 찾기 / last_name으로 찾기)

9-1-① 인덱스 생성 (first_name)

```
create index employees_idx3 on employees (first_name) ;
```

9-1-② 비효율은 없을까? (eg. first_name : Shigeu, last_name : Matzen)

```
select a.first_name||' '||a.last_name emp_name
      , ceil((sysdate - birth_date) / 365) age
      , a.hire_date
      , c.dept_name
      , d.title
      , e.salary
from employees a, dept_emp b, departments c, titles d, salaries e
where a.first_name like :v_first_name||'%'
  and a.last_name like :v_last_name||'%'
  and a.emp_no = b.emp_no and b.to_date = '99991231'
  and b.dept_no = c.dept_no
  and a.emp_no = d.emp_no and d.to_date = '99991231'
  and a.emp_no = e.emp_no and e.to_date = '99991231' ;
```

바인드 변수 (bind variable)

실습 9. UNION ALL을 활용한 실행계획 분리

9-1. 사원 이름으로 사원정보(성명, 연령, 입사일자, 소속부서명, 직급명, 급여) 검색 (first_name으로 찾기 / last_name으로 찾기)

9-1-③ 실행계획 분리 (first_name으로 검색하는 경우와 last_name으로 검색하는 경우를 분리하여 SQL 작성)

```
select /*+ ordered use_nl(a b c d e) index(a employees_idx1) */ a.first_name||' '||a.last_name emp_name
      , ceil((sysdate - birth_date) / 365) age
      , a.hire_date
      , c.dept_name
      , d.title
      , e.salary
from employees a, dept_emp b, departments c, titles d, salaries e
where :v_gubun = 1 -- last_name으로 검색
  and a.last_name like :v_last_name||'%'
  and a.emp_no = b.emp_no and b.to_date = '99991231'
  and b.dept_no = c.dept_no
  and a.emp_no = d.emp_no and d.to_date = '99991231'
  and a.emp_no = e.emp_no and e.to_date = '99991231'
union all
select /*+ ordered use_nl(a b c d e) index(a employees_idx3) */ a.first_name||' '||a.last_name emp_name
      , ceil((sysdate - birth_date) / 365) age
      , a.hire_date
      , c.dept_name
      , d.title
      , e.salary
from employees a, dept_emp b, departments c, titles d, salaries e
where :v_gubun = 2 -- first_name으로 검색
  and a.first_name like :v_first_name||'%'
  and a.emp_no = b.emp_no and b.to_date = '99991231'
  and b.dept_no = c.dept_no
  and a.emp_no = d.emp_no and d.to_date = '99991231'
  and a.emp_no = e.emp_no and e.to_date = '99991231' ;
```

last_name으로 검색하는 경우는 employees_idx1 인덱스 사용

바인드 변수를 활용한 필터 조건

first_name으로 검색하는 경우는 employees_idx3 인덱스 사용

실습 10. 부분합 / 데이터복제 / ROLLUP과 CUBE

10-1. 현재 부서별 직원들의 급여 합계와 급여 총합계 산출 (데이터복제를 활용한 부분합)

10-1-① 복제 테이블 생성

```
create table copy_t (
  no number(2) not null,
  no2 varchar2(2) not null );
```

10-1-② 복제 테이블 데이터 생성

```
insert into copy_t
select rownum
      , lpad(rownum,2,'0')
from employees
where rownum <= 99 ; -- rownum : STOP KEY

commit ;
```

ROWNUM의 이해

- **정의**
오라클에서 지원하는 가상컬럼으로 쿼리의 결과에 1부터 하나씩 증가하여 붙는 가상(pseudo) 컬럼
- **주요용도**
주로 여러개의 결과를 출력하는 쿼리문을 실행 후 결과의 개수를 제한하여 가져오는데 사용 (stop key라고 불림)
- **주의할 점**
rownum이 결과에서 1부터 순서대로 증가하여 붙기 때문에 rownum=2 나 rownum>1과 같은 방식으로는 원하는 결과를 얻을 수 있음

실습 10. 부분합 / 데이터복제 / ROLLUP과 CUBE

10-1. 현재 부서별 직원들의 급여 합계와 급여 총합계 산출 (데이터복제를 활용한 부분합)

10-1-③ 데이터복제를 통한 급여 부분합(엑셀 부분합과 유사) 산출

```
select nvl(y.dept_name, '합계') 부서명
      , x.sum_sal 급여합
from ( select decode(b.no, 1, a.dept_no, '합계') dept_no
      , sum(sum_sal) sum_sal
      from ( select b.dept_no
            , sum(a.salary) sum_sal
            from salaries a, dept_emp b
            where a.emp_no = b.emp_no
                  and a.to_date = '99991231' and b.to_date = '99991231'
            group by b.dept_no ) a, copy_t b
      where b.no <= 2
      group by no, decode(b.no, 1, a.dept_no, '합계') ) x, departments y
where x.dept_no = y.dept_no(+);
```

- copy_t 테이블과의 곱집합(cartesian product)을 통한 데이터복제
 - copy_t.no = 1 → 부서별 급여합 산출에 사용
 - copy_t.no = 2 → 전체 급여합 산출에 사용

출력 결과

부서명	급여합
1 Customer Service	1182134209
2 Development	4153249050
3 Finance	977049936
4 Human Resources	824464664
5 Marketing	1188233434
6 Production	3616319369
7 Quality Management	951919236
8 Research	1048650423
9 Sales	3349845802
10 합계	17291866123

실습 10. 부분합 / 데이터복제 / ROLLUP과 CUBE

10-2. 현재 부서별 직원들의 급여 합계와 급여 총합계 산출 (rollup() 함수를 활용한 부분합)

```
select nvl(y.dept_name, '합계') 부서명
      , x.sum_sal 급여합
from ( select b.dept_no
      , sum(a.salary) sum_sal
      from salaries a, dept_emp b
      where a.emp_no = b.emp_no
        and a.to_date = '99991231' and b.to_date = '99991231'
      group by rollup(b.dept_no) ) x, departments y
where x.dept_no = y.dept_no(+);
```

rollup() 함수 사용

앞의 데이터복제 활용결과와
동일

출력 결과

부서명	급여합
1 Customer Service	1182134209
2 Development	4153249050
3 Finance	977049936
4 Human Resources	824464664
5 Marketing	1188233434
6 Production	3616319369
7 Quality Management	951919236
8 Research	1048650423
9 Sales	3349845802
10 합계	17291866123

개념 이해 10. Rollup() / Cube() / Grouping Sets() 함수

➔ Rollup()

- ROLLUP에 지정된 Grouping Columns의 List는 Subtotal을 생성하기 위해 사용됨
- Grouping Columns의 수를 N이라고 했을 때 N+1 Level의 Subtotal이 생성됨
- $\text{Rollup}(A, B) = (\text{group by } A, B) \cup (\text{group by } A) \cup (\text{group by } \text{NULL})$

➔ Cube()

- 결합 가능한 모든 값에 대하여 다차원 집계를 생성
- Grouping Columns의 수를 N이라고 했을 때 2^N Level의 Subtotal이 생성됨
- $\text{Cube}(A, B) = (\text{group by } A, B) \cup (\text{group by } A) \cup (\text{group by } B) \cup (\text{group by } \text{NULL})$

➔ Grouping Sets()

- GROUPING SETS에 표시된 인수들에 대한 개별 집계를 구하기 위해 사용됨
- Grouping Columns의 수를 N이라고 했을 때 N Level의 Subtotal이 생성됨
- $\text{Grouping Sets}(A, B) = (\text{group by } A) \cup (\text{group by } B)$

➔ Grouping()

- ROLLUP, CUBE, GROUPING SETS 등 새로운 그룹 함수를 지원하기 위해 추가된 함수
- If $\text{expr} = (\text{ROLLUP이나 CUBE에 의한 소계가 계산된 결과})$, $\text{grouping}(\text{expr}) = 1$. if not, $\text{grouping}(\text{expr}) = 0$
- CASE/DECODE를 이용해, 소계를 나타내는 필드에 원하는 문자열을 지정할 수 있음

실습 10. 부분합 / 데이터복제 / ROLLUP과 CUBE

10-3. 현재 부서별/직급별 급여합 및 전체 급여합 산출 (cube() 함수를 활용한 부분합)

```

select nvl(y.dept_name, '합계') 부서명
      , x.title 직급명
      , x.sum_sal 급여합
from ( select decode(grouping(b.dept_no), 1, '합계', b.dept_no) dept_no
      , decode(grouping(c.title), 1, '합계', c.title) title
      , sum(a.salary) sum_sal
  from salaries a, dept_emp b, titles c
 where a.emp_no = b.emp_no
   and b.emp_no = c.emp_no
   and a.to_date = '99991231' and b.to_date = '99991231' and c.to_date = '99991231'
 group by cube(b.dept_no, c.title) ) x, departments y
where x.dept_no = y.dept_no(+);
    
```

cube() 함수 사용

출력 결과

부서명	직급명	급여합
1 Customer Service	Assistant Engineer	4013699
2 Customer Service	Technique Leader	16130639
3 Customer Service	Senior Engineer	127490191
4 Customer Service	Senior Staff	791929601
5 Customer Service	Engineer	37359981
6 Customer Service	Manager	58745
7 Customer Service	Staff	205151353
8 Customer Service	합계	1182134209
...		
55 합계	Assistant Engineer	205655454
56 합계	Technique Leader	813791946
57 합계	Senior Engineer	6086495408
58 합계	Senior Staff	6619869618
59 합계	Engineer	1846671624
60 합계	Manager	699513
61 합계	Staff	1718682560
62 합계	합계	17291866123

group by 부서, 직급

group by 부서

group by 직급

group by NULL