

조건부 논리 표현식 제어 : CASE & DECODE

[샘플 문제 1] HR 부서에서는 신규 프로젝트의 성공으로 해당하는 각 업무 자들에 대한 급여 인상안을 결정하고, 다음과 같이 업무 별 급여 인상에 대해 적용하고자 한다. 현재 107 명의 사원은 19 개의 업무에 소속되어 근무 중이다. (Distinct job\_id) 이 중 5 개의 업무 자들에 대한 급여 인상이 각각 결정되었고, 나머지 업무에 대해서는 인상이 동결되었다 (107 행). HR\_REP(10%), MK\_REP(12%), PR\_REP(15%), SA\_REP(18%), IT\_PROG(20%)

<Hint> CASE 와 DECODE 를 이용하여 위 조건을 만족하는 구문을 작성해 본다.

-- CASE

```
SELECT employee_id, last_name || ' ' || first_name as "Name", job_id, salary,

       CASE job_id WHEN 'HR_REP' THEN 1.10 * salary

                WHEN 'MK_REP' THEN 1.12 * salary

                WHEN 'PR_REP' THEN 1.15 * salary

                WHEN 'SA_REP' THEN 1.18 * salary

                WHEN 'IT_PROG' THEN 1.20 * salary

       ELSE      salary

       END      "New Salary"

FROM   Employees;
```

-- DECODE

```
SELECT employee_id, last_name || ' ' || first_name as "Name", job_id, salary,

       DECODE (job_id, 'HR_REP', 1.10 * salary

                , 'MK_REP' , 1.12 * salary

                , 'PR_REP' , 1.15 * salary

                , 'SA_REP' , 1.18 * salary

                , 'IT_PROG' , 1.20 * salary

                , salary) "New Salary"
```

FROM Employees;

[샘플 문제 2] HR 부서에서는 최상위 입사일에 해당하는 2001 년부터 2003 년까지 입사자들의 급여를 각각 5%, 3%, 1% 인상하여 지분에 따른 배당금으로 지급하고자 한다. 전체 사원들 중 해당 년도에 해당하는 사원들의 급여를 검색하여 적용하고, 입사일자에 따른 오름차순 정렬을 수행하시오.

<Hint> CASE 구문을 이용한 검색 조건 비교를 이용하여 해당 년도 별 조건 처리를 수행할 수 있다.

```
SELECT employee_id, last_name || ' ' || first_name as "Name", hire_date, salary,
       CASE
           WHEN hire_date < TO_DATE('2002-01-01', 'YYYY-MM-DD') THEN salary * 1.05
           WHEN hire_date < TO_DATE('2003-01-01', 'YYYY-MM-DD') THEN salary * 1.03
           WHEN hire_date < TO_DATE('2004-01-01', 'YYYY-MM-DD') THEN salary * 1.01
           ELSE salary
       END "New Salary"
FROM Employees
ORDER BY hire_date;
```

[문제 1] 부서별 급여 합계를 구하고, 그 결과를 가지고 다음과 같이 표현하시오(12 행).

Sum Salary > 100000 이면, "Excellent"

Sum Salary > 50000 이면, "Good"

Sum Salary > 10000 이면, "Medium"

Sum Salary <= 10000 이면, "Well"

<Hint> case 문을 사용하는 보통 방법을 사용할 수도 있고, inline view 를 이용하여 우선 부서별 급여 합계를 구하고, 상위 쿼리에서 CASE 구문을 이용하여 위의 조건 비교를 통해 급여 합계에 따른 표현을 할 수 있다.

```
SELECT department_id, SUM(salary) sum_sal,
       CASE WHEN SUM(salary) > 100000 THEN 'Excellent'
            WHEN SUM(salary) > 50000 THEN 'Good'
            WHEN SUM(salary) > 10000 THEN 'Medium'
            WHEN SUM(salary) <= 10000 THEN 'Well'
       END "Department Grade Salary"
FROM   Employees
GROUP BY department_id;
```

(인라인 뷰 사용)

```
SELECT department_id, sum_sal,
       CASE WHEN sum_sal > 100000 THEN 'Excellent'
            WHEN sum_sal > 50000 THEN 'Good'
            WHEN sum_sal > 10000 THEN 'Medium'
            WHEN sum_sal <= 10000 THEN 'Well'
       END "Department Grade Salary"
FROM   (SELECT department_id, sum(salary) sum_sal
        FROM employees
```

GROUP BY department\_id

ORDER BY department\_id);

[문제 2] 2005 년 이전에 입사한 직원 중 업무에 "MGR"이 포함된 직원은 15% 급여를 인상하고, "MAN"이 포함된 직원은 20% 급여 인상이 정해졌고, 또한 2005 년부터 근무를 시작한 직원 중 "MGR"이 포함된 직원은 25% 급여 인상을 수행하는 쿼리를 작성하시오. 해당되는 직원들만 출력한다. (7 행).

SELECT employee\_id, last\_name || ' ' || first\_name as "Name", job\_id, hire\_date, salary,

CASE

WHEN hire\_date < TO\_DATE('20050101', 'yyyymmdd')

and job\_id LIKE '%MGR%' THEN salary\*1.15

WHEN hire\_date < TO\_DATE('20050101', 'yyyymmdd')

and job\_id LIKE '%MAN%' THEN salary\*1.20

WHEN hire\_date >= TO\_DATE('20050101', 'yyyymmdd')

and job\_id LIKE '%MGR%' THEN salary\*1.25

END "Department Grade Salary"

FROM Employees

WHERE ((hire\_date < TO\_DATE('20050101', 'yyyymmdd') AND

(job\_id like '%MGR' OR job\_id like '%MAN')))

OR

((hire\_date >= TO\_DATE('20050101', 'yyyymmdd')) AND

(job\_id like '%MGR')));

(2 중 case 문)

SELECT employee\_id, last\_name || ' ' || first\_name as "Name", job\_id, hire\_date, salary,

CASE WHEN hire\_date < TO\_DATE('2005-01-01', 'YYYY-MM-DD')

THEN CASE WHEN job\_id like '%MGR' THEN salary \* 1.15

```

        WHEN job_id like '%MAN' THEN salary * 1.20

        ELSE salary END

    ELSE CASE  WHEN job_id like '%MGR' THEN salary * 1.25

        ELSE salary END

    END "Job Salary"

FROM employees

WHERE ((hire_date < TO_DATE('20050101', 'yyyymmdd') AND

        (job_id like '%MGR' OR job_id like '%MAN'))))

OR

(((hire_date >= TO_DATE('20050101', 'yyyymmdd')) AND

        (job_id like '%MGR')));

```

[문제 3] 월별로 입사한 사원 수 출력

(방식 1) 월별로 입사한 사원수가 아래와 같이 각 행별로 출력되도록 하시오(12 행).

(방식 2) 첫 행에 모든 월별 입사 사원수가 출력되도록 하시오(1 행).

<Hint> DECODE, CASE 함수는 IF THEN ELSE 로직을 구현 할 수 있도록 만든 함수이다. 기본 IF 구문의 흐름처럼 조건에 맞는 행은 각 DECODE 절이 count 하고, 아닌 것은 0 으로 처리하는 IF 구문을 작성하는 것과 유사하다. 1~12 월까지의 IF 구문이 있다고 보고, 각 IF 문 즉, 각 DECODE 함수의 합을 구하면 된다.

방식 1)

(case 사용)

```

SELECT

    CASE WHEN TO_CHAR(hire_date, 'MM') = '01'

        THEN count(*) ELSE 0 END "1 Month",

    CASE WHEN TO_CHAR(hire_date, 'MM') = '02'

        THEN count(*) ELSE 0 END "2 Month",

```

```

CASE WHEN TO_CHAR(hire_date, 'MM') = '03'
      THEN count(*) ELSE 0 END "3 Month",
CASE WHEN TO_CHAR(hire_date, 'MM') = '04'
      THEN count(*) ELSE 0 END "4 Month",
CASE WHEN TO_CHAR(hire_date, 'MM') = '05'
      THEN count(*) ELSE 0 END "5 Month",
CASE WHEN TO_CHAR(hire_date, 'MM') = '06'
      THEN count(*) ELSE 0 END "6 Month",
CASE WHEN TO_CHAR(hire_date, 'MM') = '07'
      THEN count(*) ELSE 0 END "7 Month",
CASE WHEN TO_CHAR(hire_date, 'MM') = '08'
      THEN count(*) ELSE 0 END "8 Month",
CASE WHEN TO_CHAR(hire_date, 'MM') = '09'
      THEN count(*) ELSE 0 END "9 Month",
CASE WHEN TO_CHAR(hire_date, 'MM') = '10'
      THEN count(*) ELSE 0 END "10 Month",
CASE WHEN TO_CHAR(hire_date, 'MM') = '11'
      THEN count(*) ELSE 0 END "11 Month",
CASE WHEN TO_CHAR(hire_date, 'MM') = '12'
      THEN count(*) ELSE 0 END "12 Month"

```

FROM Employees

GROUP BY TO\_CHAR(hire\_date, 'MM')

ORDER BY TO\_CHAR(hire\_date, 'MM');

(decode 사용)

```

SELECT DECODE(TO_CHAR(hire_date, 'MM'), '01', COUNT(*), 0) "1 Month",

```

```

DECODE(TO_CHAR(hire_date,'MM'), '02', COUNT(*), 0) "2 Month",
DECODE(TO_CHAR(hire_date,'MM'), '03', COUNT(*), 0) "3 Month",
DECODE(TO_CHAR(hire_date,'MM'), '04', COUNT(*), 0) "4 Month",
DECODE(TO_CHAR(hire_date,'MM'), '05', COUNT(*), 0) "5 Month",
DECODE(TO_CHAR(hire_date,'MM'), '06', COUNT(*), 0) "6 Month",
DECODE(TO_CHAR(hire_date,'MM'), '07', COUNT(*), 0) "7 Month",
DECODE(TO_CHAR(hire_date,'MM'), '08', COUNT(*), 0) "8 Month",
DECODE(TO_CHAR(hire_date,'MM'), '09', COUNT(*), 0) "9 Month",
DECODE(TO_CHAR(hire_date,'MM'), '10', COUNT(*), 0) "10 Month",
DECODE(TO_CHAR(hire_date,'MM'), '11', COUNT(*), 0) "11 Month",
DECODE(TO_CHAR(hire_date,'MM'), '12', COUNT(*), 0) "12 Month"

```

FROM Employees

GROUP BY TO\_CHAR(hire\_date,'MM')

ORDER BY TO\_CHAR(hire\_date,'MM');

방식 2)

(case 사용)

SELECT

```

SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '01'
      THEN count(*) ELSE 0 END) "1 Month",
SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '02'
      THEN count(*) ELSE 0 END) "2 Month",
SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '03'
      THEN count(*) ELSE 0 END) "3 Month",
SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '04'
      THEN count(*) ELSE 0 END) "4 Month",

```

```

SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '05'
      THEN count(*) ELSE 0 END) "5 Month",
SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '06'
      THEN count(*) ELSE 0 END) "6 Month",
SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '07'
      THEN count(*) ELSE 0 END) "7 Month",
SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '08'
      THEN count(*) ELSE 0 END) "8 Month",
SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '09'
      THEN count(*) ELSE 0 END) "9 Month",
SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '10'
      THEN count(*) ELSE 0 END) "10 Month",
SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '11'
      THEN count(*) ELSE 0 END) "11 Month",
SUM(CASE WHEN TO_CHAR(hire_date, 'MM') = '12'
      THEN count(*) ELSE 0 END) "12 Month"

```

FROM Employees

GROUP BY TO\_CHAR(hire\_date, 'MM');

(decode 사용)

```

SELECT SUM(DECODE(TO_CHAR(hire_date,'MM'), '01', COUNT(*), 0)) "1 Month",
       SUM(DECODE(TO_CHAR(hire_date,'MM'), '02', COUNT(*), 0)) "2 Month",
       SUM(DECODE(TO_CHAR(hire_date,'MM'), '03', COUNT(*), 0)) "3 Month",
       SUM(DECODE(TO_CHAR(hire_date,'MM'), '04', COUNT(*), 0)) "4 Month",
       SUM(DECODE(TO_CHAR(hire_date,'MM'), '05', COUNT(*), 0)) "5 Month",
       SUM(DECODE(TO_CHAR(hire_date,'MM'), '06', COUNT(*), 0)) "6 Month",

```



```
SUM(DECODE(TO_CHAR(hire_date,'MM'), '07', COUNT(*), 0)) "7 Month",  
SUM(DECODE(TO_CHAR(hire_date,'MM'), '08', COUNT(*), 0)) "8 Month",  
SUM(DECODE(TO_CHAR(hire_date,'MM'), '09', COUNT(*), 0)) "9 Month",  
SUM(DECODE(TO_CHAR(hire_date,'MM'), '10', COUNT(*), 0)) "10 Month",  
SUM(DECODE(TO_CHAR(hire_date,'MM'), '11', COUNT(*), 0)) "11 Month",  
SUM(DECODE(TO_CHAR(hire_date,'MM'), '12', COUNT(*), 0)) "12 Month"
```

```
FROM Employees
```

```
GROUP BY TO_CHAR(hire_date,'MM')
```

```
ORDER BY TO_CHAR(hire_date,'MM');
```