

# 분석함수 (WINDOW)

## 분석함수란?

데이터를 분석하는 함수이다. 분석함수를 사용하면, 쿼리 실행의 결과인 RESULT SET을 대상으로 **전체 그룹별이 아닌 소그룹별로 각 행에 대한 계산값을 리턴**한다.(분석함수용 그룹을 별도로 정의하여 그룹별 계산)

## 표현식

```
분석함수명 ([전달인자1[, 전달인자2[, 전달인자3]])  
OVER ([PARTITION BY 절]  
      [ORDER BY 절]  
      [WINDOW 절])
```

\*\* 전달인자는 0~3개까지 값을 지정할 수 있음

\*\* PARTITION BY 절 : 분석함수의 계산 대상 그룹을 지정하는 절

ORDER BY 절 : 계산 대상 그룹에 대한 정렬작업을 수행함

WINDOW절 : PARTITION BY절에 의해 나뉜 기준에 또 다른 소그룹을 만듦

# 분석함수(WINDOW)

## RANK 함수

등수를 매기거나 등수를 알아내는 함수로, 같은 등수가 있는 경우 다음 등수 값을 건너 뛴다.

```
SELECT EMP_ID,  
       EMP, NAME,  
       SALARY,  
       RANK () OVER (ORDER BY SALARY DESC) 급여순위  
FROM EMPLOYEE;
```

EMP_ID	EMP_NAME	SALARY	급여순위
1 200	선동일	8000000	1
2 201	송종기	6000000	2
3 205	정중하	3900000	3
4 215	대복훈	3760000	4
5 202	노용철	3700000	5
6 217	전지연	3660000	6
7 209	심봉선	3500000	7
8 204	유재식	3400000	8
9 905	길성춘	3000000	9
10 218	미오리	2890000	10
11 203	송은희	2800000	11
12 216	차태연	2780000	12
13 212	장프위	2550000	13
14 208	김해솔	2500000	14
15 220	이중석	2490000	15
16 221	유하진	2480000	16
17 222	이태림	2436240	17
18 213	하동운	2320000	18
19 207	하이유	2200000	19
20 211	전형돈	2000000	20
21 210	윤은혜	2000000	20
22 206	박나라	1800000	22
23 219	임시환	1550000	23
24 214	방명수	1380000	24

# 분석함수(WINDOW)

## RANK 함수

[지정하는 값이 전체 값 중에서 몇위인지 조회하려는 경우]

```
SELECT RANK(2300000) WITHIN GROUP (ORDER BY SALARY DESC) 순위  
FROM EMPLOYEE;
```

	순위
1	19

## DENSE\_RANK 함수

```
SELECT EMP_ID,  
       EMP, NAME,  
       SALARY,  
       DENSE_RANK ()  
         OVER (ORDER BY SALARY DESC) 급여순위  
FROM EMPLOYEE;
```

EMP_ID	EMP_NAME	SALARY	급여순위
1 200	선동일	8000000	1
2 201	송종기	6000000	2
3 205	정중하	3900000	3
4 215	대복훈	3760000	4
5 202	노용철	3700000	5
6 217	전지연	3660000	6
7 209	심봉선	3500000	7
8 204	유재식	3400000	8
9 905	길성훈	3000000	9
10 218	미오리	2890000	10
11 203	송은희	2800000	11
12 216	차태연	2780000	12
13 212	장프위	2550000	13
14 208	김해솔	2500000	14
15 220	이종석	2490000	15
16 221	유하진	2480000	16
17 222	이태림	2436240	17
18 213	하동운	2320000	18
19 207	하미유	2200000	19
20 211	전형돈	2000000	20
21 210	윤은혜	2000000	20
22 206	박나라	1800000	21
23 219	임시환	1550000	22
24 214	방명수	1380000	23

Rank = 1, 1, 3 등 (공동 1등)

Dense\_Rank = 1, 1, 2 등

# 분석함수(WINDOW)

## CUME\_DIST 함수

PARTITION BY에 의해 나누어진 그룹별로 각 행을 ORDER BY절에 명시된 순서로 정렬 후 그룹별로 **누적된 분산정도를 구하는 함수**이다.(누적분산)

```
SELECT EMP_ID,  
       EMP_NAME,  
       SALARY,  
       ROUND(CUME_DIST() OVER(ORDER BY SALARY),2) 누적분산  
FROM EMPLOYEE  
WHERE DEPT_CODE = 'D5';
```

\*\* 누적분산은  $0 < \text{결과값} \leq 1$ 의 범위이다.

	EMP_NAME	SALARY	누적분산
1	박나라	1800000	0.14
2	윤은해	2000000	0.29
3	하미유	2200000	0.43
4	김해솔	2500000	0.57
5	길성춘	3000000	0.71
6	심봉선	3500000	0.86
7	대복훈	3760000	1

# 분석함수(WINDOW)

## PERCENT\_RANK 함수

PARTITION BY에 의해 나누어진 그룹별로 각 행을 ORDER BY절에 명시된 순서로 정렬 후 그룹별로 백분위를 반환하는 함수이다.

```
SELECT EMP_ID,  
       EMP_NAME,  
       SALARY,  
       ROUND(PERCENT_RANK () OVER(ORDER BY SALARY) ,2) 백분위  
FROM EMPLOYEE  
WHERE DEPT_CODE = 'D5';
```

\*\* 백분위는  $0 \leq \text{결과값} \leq 1$ 의 범위이다.

	EMP_NAME	SALARY	백분위
1	박나라	1800000	0
2	윤은해	2000000	0.17
3	하미유	2200000	0.33
4	김해솔	2500000	0.5
5	길성춘	3000000	0.67
6	심봉선	3500000	0.83
7	대북훈	3760000	1

# 분석함수(WINDOW)

## NTILE 함수

PARTITION BY를 **BUCKET**이라고 불리는 **그룹별로 나누고 각 행을 버킷에 배치하는 함수**이다.

```
SELECT EMP_NAME,  
       SALARY,  
       NTILE(4) OVER (ORDER BY SALARY DESC) 등급  
FROM EMPLOYEE;
```

	EMP_NAME	SALARY	등급
1	선동일	8000000	1
2	송종기	6000000	1
3	정중하	3900000	1
4	대복혼	3760000	1
5	노용철	3700000	1
6	전지연	3660000	1
7	심봉선	3500000	2
8	유재식	3400000	2
9	길성춘	3000000	2
10	이오리	2890000	2
11	송은희	2800000	2
12	차태연	2780000	2
13	장프위	2550000	3
14	김해술	2500000	3
15	이종석	2490000	3
16	유하진	2480000	3
17	이태림	2436240	3
18	하동운	2320000	3
19	하미유	2200000	4
20	전철돈	2000000	4
21	윤은혜	2000000	4
22	박나라	1800000	4
23	임시환	1550000	4
24	방명수	1380000	4

\*\* 6명씩 4개 그룹으로 나누어 등급을 매김

# 분석함수(WINDOW)

## ROW\_NUMBER 함수

PARTITION BY별로 ORDER BY된 행의 순위를 반환하는 함수이다.

```
SELECT EMP_ID,  
       EMP_NAME,  
       DEPT_CODE,  
       HIRE_DATE,  
       ROW_NUMBER() OVER (PARTITION BY DEPT_CODE  
                           ORDER BY HIRE_DATE) 순번  
FROM EMPLOYEE;
```

EMP_ID	EMP_NAME	DEPT_CODE	HIRE_DATE	순번
1 217	전지연	D1	07/03/20	1
2 214	방명수	D1	10/04/04	2
3 216	차태연	D1	13/03/01	3
4 221	유하진	D2	94/01/20	1
5 219	임시환	D2	99/09/09	2
6 220	이종석	D2	14/09/18	3
7 207	하미유	D5	94/07/07	1
8 210	윤은혜	D5	01/02/03	2
9 208	김해솔	D5	04/04/30	3
10 206	박나라	D5	08/04/02	4
11 209	심봉선	D5	11/11/11	5
12 215	대복존	D5	17/06/19	6
13 905	길성춘	D5	17/09/27	7
14 203	송은희	D6	96/05/03	1
15 205	정중하	D6	99/09/09	2
16 204	유재식	D6	00/12/29	3
17 222	이태림	D8	97/09/12	1
18 211	전형돈	D8	12/12/12	2
19 212	장프위	D8	15/06/17	3
20 200	선동일	D9	90/02/06	1
21 202	노웅철	D9	01/01/01	2
22 201	송증기	D9	01/09/01	3
23 213	하동운	(null)	99/12/31	1
24 218	미오리	(null)	16/11/28	2



# 분석함수(WINDOW)

## RATIO\_TO\_REPORT 함수

PARTITION BY별로 합에 대한 비율을 반환하는 함수이다.(점유율)

```
SELECT DEPT_CODE,  
       COUNT(*) 부서인원,  
       ROUND((RATIO_TO_REPORT(COUNT(*)) OVER() * 100),1) || '%' 인원비율,  
       SUM(SALARY) 급여합계,  
       ROUND((RATIO_TO_REPORT(SUM(SALARY)) OVER() * 100),1) || '%' 급여비율  
FROM EMPLOYEE  
GROUP BY DEPT_CODE;
```

	DEPT_CODE	부서인원	인원비율	급여합계	급여비율
1	(null)	2	8.3%	5210000	7.1%
2	D1	3	12.5%	7820000	10.7%
3	D9	3	12.5%	17700000	24.2%
4	D5	7	29.2%	18760000	25.7%
5	D6	3	12.5%	10100000	13.8%
6	D2	3	12.5%	6520000	8.9%
7	D8	3	12.5%	6986240	9.6%

# 분석함수(WINDOW)

## RATIO\_TO\_REPORT 함수

[인원 대비 급여 지급 효율이 좋지 않은 부서를 조회하세요.]

```
SELECT *  
FROM (SELECT DEPT_CODE,  
             COUNT(*) 부서인원,  
             ROUND((RATIO_TO_REPORT(COUNT(*)) OVER() * 100),1) 인원비율,  
             SUM(SALARY) 급여합계,  
             ROUND((RATIO_TO_REPORT(SUM(SALARY)) OVER() * 100),1) 급여비율  
FROM EMPLOYEE  
GROUP BY DEPT_CODE)  
WHERE 급여비율 > 인원비율;
```

	DEPT_CODE	부서인원	인원비율	급여합계	급여비율
1	D9	3	12.5	17700000	24.2
2	D6	3	12.5	10100000	13.8

# 분석함수(WINDOW)

## WINDOW절 - ROWS

PARTITION BY에 의한 소그룹의 크기를 물리적 단위로 행 집합을 지정한 단위이다.

UNBOUNDED PRECEDING : 윈도우의 첫 행을 의미한다.

UNBOUNDED FOLLOWING : 윈도우의 마지막 행을 의미한다.

CURRENT ROW : 값이 기록된 현재 행을 의미한다.

# 분석함수(WINDOW)

## WINDOW절 - ROWS

```
SELECT EMP_ID, SALARY,  
       SUM(SALARY) OVER (PARTITION BY DEPT_CODE  
                          ORDER BY EMP_ID  
                          ROWS BETWEEN UNBOUNDED PRECEDING  
                          AND UNBOUNDED FOLLOWING) "WIN1",  
       SUM(SALARY) OVER (PARTITION BY DEPT_CODE  
                          ORDER BY EMP_ID  
                          ROWS BETWEEN UNBOUNDED PRECEDING  
                          AND CURRENT ROW) "WIN2",  
       SUM(SALARY) OVER (PARTITION BY DEPT_CODE  
                          ORDER BY EMP_ID  
                          ROWS BETWEEN CURRENT ROW  
                          AND UNBOUNDED FOLLOWING) "WIN3"  
FROM EMPLOYEE  
WHERE DEPT_CODE = 'D5';
```

	EMP_ID	SALARY	WIN1	WIN2	WIN3
1	206	1800000	18760000	1800000	18760000
2	207	2200000	18760000	4000000	16960000
3	208	2500000	18760000	6500000	14760000
4	209	3500000	18760000	10000000	12260000
5	210	2000000	18760000	12000000	8760000
6	215	3760000	18760000	15760000	6760000
7	905	3000000	18760000	18760000	3000000

# 분석함수(WINDOW)

## WINDOW절 - ROWS

```
SELECT EMP_ID, SALARY,  
       SUM(SALARY) OVER (PARTITION BY DEPT_CODE  
                           ORDER BY EMP_ID  
                           ROWS BETWEEN 1 PRECEDING  
                           AND 1 FOLLOWING) "WIN1",  
       SUM(SALARY) OVER (PARTITION BY DEPT_CODE  
                           ORDER BY EMP_ID  
                           ROWS BETWEEN 1 PRECEDING  
                           AND CURRENT ROW) "WIN2",  
       SUM(SALARY) OVER (PARTITION BY DEPT_CODE  
                           ORDER BY EMP_ID  
                           ROWS BETWEEN CURRENT ROW  
                           AND 1 FOLLOWING) "WIN3"  
FROM EMPLOYEE  
WHERE DEPT_CODE = 'D5';
```

	EMP_ID	SALARY	WIN1	WIN2	WIN3
1	206	1800000	4000000	1800000	4000000
2	207	2200000	6500000	4000000	4700000
3	208	2500000	8200000	4700000	6000000
4	209	3500000	8000000	6000000	5500000
5	210	2000000	9260000	5500000	5760000
6	215	3760000	8760000	5760000	6760000
7	905	3000000	6760000	6760000	3000000

# 분석함수(WINDOW)

## LAG 함수

```
SELECT EMP_NAME, DEPT_CODE, SALARY,  
       LAG(SALARY, 1, 0) OVER (ORDER BY DEPT_CODE, EMP_NAME) 이전값,  
       LAG(SALARY, 1, SALARY) OVER (ORDER BY DEPT_CODE, EMP_NAME) "조회2",  
       LAG(SALARY, 1, SALARY) OVER (PARTITION BY DEPT_CODE  
                                   ORDER BY DEPT_CODE, EMP_NAME) "조회3"  
FROM EMPLOYEE;
```

EMP_NAME	DEPT_CODE	SALARY	이전값	조회2	조회3
1 방명수	D1	1380000	0	1380000	1380000
2 전지연	D1	3660000	1380000	1380000	1380000
3 차태연	D1	2780000	3660000	3660000	3660000
4 유하진	D2	2480000	2780000	2780000	2480000
5 이종석	D2	2490000	2480000	2480000	2480000
6 임시환	D2	1550000	2490000	2490000	2490000
7 길성춘	D5	3000000	1550000	1550000	3000000
8 김해술	D5	2500000	3000000	3000000	3000000
9 대복훈	D5	3760000	2500000	2500000	2500000
10 박나라	D5	1800000	3760000	3760000	3760000
11 심봉선	D5	3500000	1800000	1800000	1800000
12 윤은혜	D5	2000000	3500000	3500000	3500000
13 하미유	D5	2200000	2000000	2000000	2000000
14 송은희	D6	2800000	2200000	2200000	2800000
15 유재식	D6	3400000	2800000	2800000	2800000
16 정중하	D6	3900000	3400000	3400000	3400000
17 이태림	D8	2436240	3900000	3900000	2436240
18 장프위	D8	2550000	2436240	2436240	2436240
19 전철도	D8	2000000	2550000	2550000	2550000
20 노용철	D9	3700000	2000000	2000000	3700000
21 선동일	D9	8000000	3700000	3700000	3700000
22 송종기	D9	6000000	8000000	8000000	8000000
23 미오리	(null)	2890000	6000000	6000000	2890000
24 하동운	(null)	2320000	2890000	2890000	2890000

# 분석함수(WINDOW)

## LEAD 함수

```
SELECT EMP_NAME, DEPT_CODE, SALARY,  
       LEAD(SALARY, 1, 0) OVER (ORDER BY DEPT_CODE, EMP_NAME) 다음값,  
       LEAD(SALARY, 1, SALARY) OVER (ORDER BY DEPT_CODE, EMP_NAME) "조회2",  
       LEAD(SALARY, 1, SALARY) OVER (PARTITION BY DEPT_CODE  
                                     ORDER BY DEPT_CODE, EMP_NAME) "조회3"  
FROM EMPLOYEE;
```

EMP_NAME	DEPT_CODE	SALARY	다음값	조회2	조회3
1 방명수	D1	1380000	3660000	3660000	3660000
2 전지연	D1	3660000	2780000	2780000	2780000
3 차태연	D1	2780000	2480000	2480000	2780000
4 유하진	D2	2480000	2490000	2490000	2490000
5 이종석	D2	2490000	1550000	1550000	1550000
6 임시환	D2	1550000	3000000	3000000	1550000
7 길성훈	D5	3000000	2500000	2500000	2500000
8 김해솔	D5	2500000	3760000	3760000	3760000
9 대복혼	D5	3760000	1800000	1800000	1800000
10 박나라	D5	1800000	3500000	3500000	3500000
11 심봉선	D5	3500000	2000000	2000000	2000000
12 윤은혜	D5	2000000	2200000	2200000	2200000
13 하미유	D5	2200000	2800000	2800000	2200000
14 송은희	D6	2800000	3400000	3400000	3400000
15 유재식	D6	3400000	3900000	3900000	3900000
16 정중하	D6	3900000	2436240	2436240	3900000
17 이태림	D8	2436240	2550000	2550000	2550000
18 장프위	D8	2550000	2000000	2000000	2000000
19 전형돈	D8	2000000	3700000	3700000	2000000
20 노용철	D9	3700000	8000000	8000000	8000000
21 선동일	D9	8000000	6000000	6000000	6000000
22 송종기	D9	6000000	2890000	2890000	6000000
23 미오리	(null)	2890000	2320000	2320000	2320000
24 하동운	(null)	2320000	0	2320000	2320000