



# SQL 활용

## 순위 계산



한국기술교육대학교  
온라인평생교육원

## 학습내용

- 순위 함수
- 그룹 별 순위

## 학습목표

- 질의 결과에 순위를 지정하는 순위 함수를 사용하여 순위를 추출할 수 있다.
- 순위 함수를 적용하여 그룹별 순위를 부여할 수 있다.

## ● 순위 함수

### 1. TOP() 함수

#### ◆ 질의 결과 튜플 수의 제한

- 질의 결과는 ORDER BY 절을 이용하여 정렬할 수 있음
- ORDER BY 정렬 기준에서 특정 등수 / 비율까지만 보고 싶은 경우

⇒ Top(n) 함수를 이용함

```
SELECT TOP(n) 속성명  
...  
ORDER BY 속성명
```

Q 사원들 중 급여 기준 5등까지만 결과로 출력하기

The screenshot shows a SQL query in the Enterprise Manager query window:

```
USE MagicCorp  
GO  
select top(5) *  
from employee  
order by salary desc
```

Below the query window, the 'Results' tab is active, displaying a table with 9 columns: ENO, ENAME, JOB, MANAGER, HIREDATE, SALARY, COMMISSION, and DNO. The table contains 5 rows of data, ordered by salary in descending order.

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	109	e9	ceo	NULL	1996-10-04 00:00:00.000	1000	NULL	20
2	104	e4	chief	1008	2003-09-02 00:00:00.000	600	NULL	20
3	113	e13	chief	1003	2002-10-09 00:00:00.000	560	NULL	20
4	107	e7	chief	1008	2004-01-08 00:00:00.000	520	NULL	10
5	103	e3	section	1005	2005-02-10 00:00:00.000	500	100	30

- 5등에 동물이 있을 경우 임의로 한 개만 출력함

## ● 순위 함수

### 1. TOP() 함수

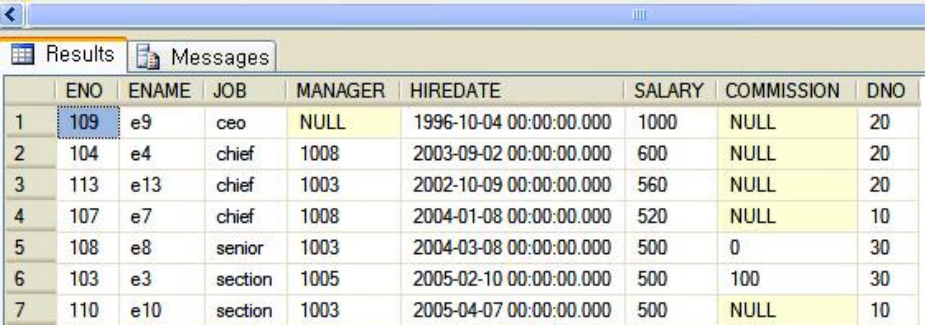
#### ◆ 동률이 있을 때 모두 보고 싶은 경우

- WITH TIES를 사용함

```
SELECT TOP(n) WITH TIES 속성명  
...  
ORDER BY 속성명
```

#### Q 사원들 중 급여 기준 5등까지만 결과로 출력하기

```
USE MagicCorp  
GO  
select top(5) WITH TIES *  
from employee  
order by salary desc
```



	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	109	e9	ceo	NULL	1996-10-04 00:00:00.000	1000	NULL	20
2	104	e4	chief	1008	2003-09-02 00:00:00.000	600	NULL	20
3	113	e13	chief	1003	2002-10-09 00:00:00.000	560	NULL	20
4	107	e7	chief	1008	2004-01-08 00:00:00.000	520	NULL	10
5	108	e8	senior	1003	2004-03-08 00:00:00.000	500	0	30
6	103	e3	section	1005	2005-02-10 00:00:00.000	500	100	30
7	110	e10	section	1003	2005-04-07 00:00:00.000	500	NULL	10

- 5등에 동률이 있을 경우 모두 출력함

## ● 순위 함수

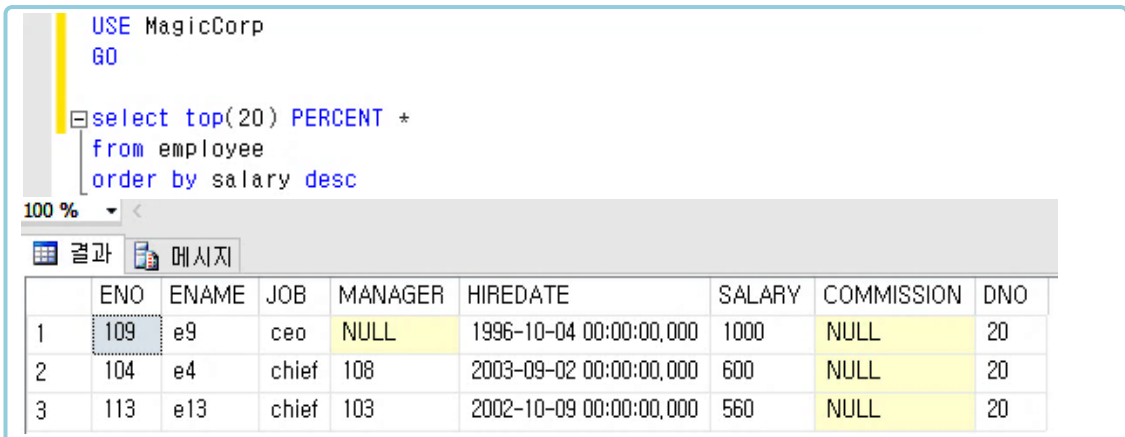
### 1. TOP() 함수

#### ◆ 정렬 기준 특정 비율까지만 보고 싶은 경우

- Top(n) PERCENT를 이용함
- 상위 n%까지만을 출력하게 됨
  - WITH TIES와도 같이 쓸 수 있음

```
SELECT TOP(n) PERCENT [WITH TIES] 속성명  
...  
ORDER BY 속성명
```

Q 사원들 중 급여 기준 20%까지만 결과로 출력하기



The screenshot shows a SQL query in the Enterprise Manager interface. The query is:

```
USE MagicCorp  
GO  
  
select top(20) PERCENT *  
from employee  
order by salary desc
```

Below the query, the results are displayed in a table. The table has columns: ENO, ENAME, JOB, MANAGER, HIREDATE, SALARY, COMMISSION, and DNO. The results are ordered by salary in descending order, showing the top 20% of employees.

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	109	e9	ceo	NULL	1996-10-04 00:00:00,000	1000	NULL	20
2	104	e4	chief	108	2003-09-02 00:00:00,000	600	NULL	20
3	113	e13	chief	103	2002-10-09 00:00:00,000	560	NULL	20

## ● 순위 함수

### 2. RANK() 함수

TOP() 함수를 쓰면 결과 수를 제한함

- TOP() 함수는 등수를 구할 수 없음

#### ◆ RANK 함수

- 각 튜플에 등수를 표시함

RANK 함수 over (order by 속성명 [asc|desc] )

- 속성 기준 오름차순(asc)또는 내림차순(desc)으로 정렬된 상태에 대하여 등수 지정
- 다양함 RANK 함수가 있음

#### ◆ RANK() 함수

SELECT 속성명, RANK () OVER (ORDER BY 속성명 [asc|desc] )

- 동률에 대하여 동일 등수 배정
- 비연속식 등수 배정

예) 1,2,2,4,...

Q 사원에 대하여 이름, 급여, 급여에 대한 내림차순 RANK() 값 출력하기

```
USE MagicCorp
GO

select ename, salary, rank() over(order by salary desc) as rank
from employee
```

100 %

결과 메시지

	ename	salary	rank
1	e9	1000	1
2	e4	600	2
3	e13	560	3
4	e7	520	4
5	e8	500	5
6	e3	500	5
7	e10	500	5
8	e6	480	8
9	e5	450	9

## ● 순위 함수

### 2. RANK() 함수

#### ◆ DENSE\_RANK() 함수

```
SELECT 속성명, DENSE_RANK () OVER (ORDER BY 속성명)
```

- 동렬에 대하여 동일 등수 배정
- 연속식 등수 배정

예) 1,2,2,3,...

Q 사원에 대하여 이름, 급여, 급여에 대한 DENSE\_RANK() 값 출력하기

```
USE MagicCorp
GO

select ename, salary, dense_rank() over(order by salary desc) as rank
from employee
```

100 %

결과 메시지

	ename	salary	rank
1	e9	1000	1
2	e4	600	2
3	e13	560	3
4	e7	520	4
5	e8	500	5
6	e3	500	5
7	e10	500	5
8	e6	480	6
9	e5	450	7

## ● 순위 함수

### 2. RANK() 함수

#### ◆ ROW\_NUMBER() 함수

```
SELECT 속성명, ROW_NUMBER () OVER (ORDER BY 속성명)
```

- 동렬에 대하여 임의 등수 배정
- 연속식 등수 배정

예) 1,2,3,4,...

Q 사원에 대하여 이름, 급여, 급여에 대한 ROW\_NUMBER() 값 출력하기

```
USE MagicCorp  
GO
```

```
select ename, salary, ROW_NUMBER() over(order by salary desc) as rank  
from employee
```

100 %			
결과 메시지			
	ename	salary	rank
1	e9	1000	1
2	e4	600	2
3	e13	560	3
4	e7	520	4
5	e8	500	5
6	e3	500	6
7	e10	500	7
8	e6	480	8
9	e5	450	9



## ● 순위 함수

### 2. RANK() 함수

#### ◆ NTILE(n) 함수

- 전체 튜플을 n개로 균등 분할하여 순위 지정

- 예**
- 결과 튜플이 20개이고 n이 10이면, 1등 2개, 2등 2개, ..., 10등 2개로 등수 지정
  - 결과 튜플수가 n으로 나누어 떨어지지 않으면 1등부터 추가적으로 배정함
  - 결과 튜플이 22개이고 n이 10이면 1등 3개, 2등 3개, 3등 2개, ..., 10등 2개로 등수 지정

**Q** 사원에 대하여 이름, 급여, 급여에 대한 내림차순으로 5등분 하여 등분순위 출력하기

```
USE MagicCorp
GO

select ename, salary, NTILE(5) over(order by salary desc) as rank
from employee
```

100 %

결과 메시지

	ename	salary	rank		ename	salary	rank
1	e9	1000	1	8	e6	480	3
2	e4	600	1	9	e5	450	3
3	e13	560	1	10	e1	300	4
4	e7	520	2	11	e12	300	4
5	e8	500	2	12	e11	280	4
6	e3	500	2	13	e14	250	5
7	e10	500	3	14	e2	250	5

## ● 그룹 별 순위

### 1. 그룹 별 순위 지정

#### ◆ 기존 RANK() 함수 문법

- 전체 결과에 대한 속성값 기준 등수 지정이 됨
  - 특정 그룹별 순위 지정은 어떻게 할까?
    - 부서별로 구분해서 각 부서 내에서 봉급 순위를 알아봄

#### ◆ PARTITION BY 속성명

- 튜플들을 속성값에 따라서 그룹핑함
- 각 그룹에 대하여 순위 함수를 적용함

```
RANK() over (PARTITION BY dno ORDER BY salary desc)
```

- “DNO별로 분류하고 각 분류된 소그룹에서 salary기준 내림차순하고 순위를 나타내시오.”라는 의미임

**Q** DNO별로 분류하고 각 분류된 소그룹에서 salary기준 내림차순하고 순위 나타내기

```
USE MagicCorp
GO

select ename, salary, dno,
       rank() over(partition by dno order by salary desc) as rank_dept
from employee
```

	ename	salary	dno	rank_dept		ename	salary	dno	rank_dept
1	e7	520	10	1	8	e1	300	20	4
2	e10	500	10	2	9	e8	500	30	1
3	e14	250	10	3	10	e3	500	30	1
4	e9	1000	20	1	11	e6	480	30	3
5	e4	600	20	2	12	e5	450	30	4
6	e13	560	20	3	13	e11	280	30	5
7	e12	300	20	4	14	e2	250	30	6

## ● 그룹 별 순위

### 1. 그룹 별 순위 지정

#### ◆ 그룹별 특정 등수의 정보를 보고 싶은 경우

- WHERE 절을 같이 활용함

```
RANK() over (PARTITION BY dno ORDER BY salary desc) AS 속성명  
...
```

```
WHERE 속성명 = 등수
```

**Q** 각 부서에서 급여 순위 2등인 사원의 부서번호, 이름과 급여 출력하기

- 인라인 뷰를 사용함

```
USE MagicCorp
GO

select *
FROM
  (SELECT dno, ename, salary, rank() over(partition by dno order by salary desc) as rank_val
   FROM EMPLOYEE) as temp
WHERE rank_val =2
```

100 % <

	dno	ename	salary	rank_val
1	10	e10	500	2
2	20	e4	600	2

## ● 그룹 별 순위

### 2. 그룹 별 집단 함수

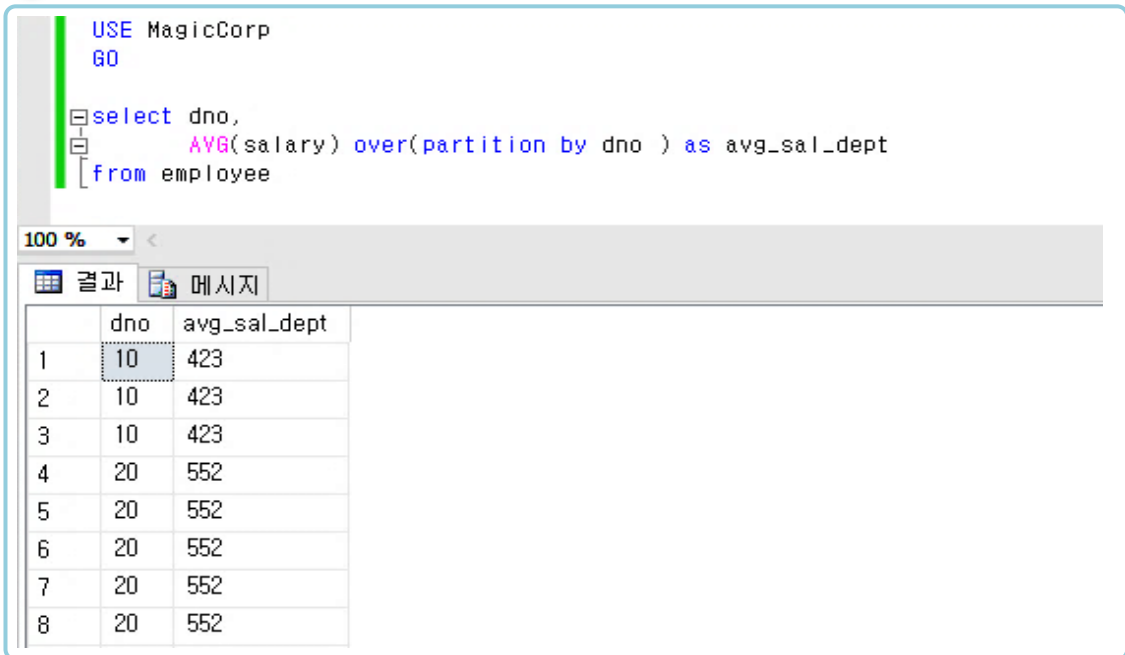
#### ◆ 그룹 별 집단 함수의 적용

SELECT 집단 함수 ~ GROUP BY~

#### ◆ PARTITION BY를 이용해서도 그룹 별 집단 함수를 적용할 수 있음

SELECT 집단함수() OVER (PARTITION BY 속성명)

#### Q PARTITION BY를 이용하여 부서별 급여의 평균 출력하기



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

```
USE MagicCorp
GO

select dno,
       avg(salary) over(partition by dno ) as avg_sal_dept
from employee
```

Below the query editor, the '결과' (Results) tab is active, displaying the following data:

	dno	avg_sal_dept
1	10	423
2	10	423
3	10	423
4	20	552
5	20	552
6	20	552
7	20	552
8	20	552

## ● 그룹 별 순위

### 3. 행 순서 함수

#### ◆ 행 순서 함수란?

- 행 순서 함수 : 정렬된 대상에서 특정 순위의 튜플들을 추출할 필요가 있을 때 사용되는 함수
  - FIRST\_VALUE 함수
    - 정렬 대상에서 첫 번째 데이터 추출
  - LAG / LEAD 함수
    - 지정된 순서에서 선행 / 후행 데이터를 참조하는 함수

#### ◆ FIRST\_VALUE 함수

##### Q 각 부서별 최고 급여액 출력하기

```
USE MagicCorp
GO

select DISTINCT dno, FIRST_VALUE(salary)
over(partition by dno order by salary desc) as highest_sal
from employee
```

100 % <

결과 메시지

	dno	highest_sal
1	10	520
2	20	1000
3	30	500

## ● 그룹 별 순위

### 3. 행 순서 함수

#### ◆ LAG / LEAD 함수

- 정렬 기준 선행 값 / 후행 값을 추출함

**Q** 각 사원별 이름, 급여와 급여 순위 상 선행 순위의 급여, 급여 순위 상 후행 순위의 급여 출력하기

```
USE MagicCorp
GO
```

```
select DISTINCT ename, salary,
                LAG(salary, 1) over(order by salary desc) as LAG_VAL,
                LEAD(salary, 1) over(order by salary desc) as LEAD_VAL
from employee
```

100 %

결과 메시지

	ename	salary	LAG_VAL	LEAD_VAL
1	e1	300	450	300
2	e10	500	500	480
3	e11	280	300	250
4	e12	300	300	280
5	e13	560	600	520
6	e14	250	280	250
7	e2	250	250	NULL
8	e3	500	500	500
9	e4	600	1000	560

## 핵심요약

### 1. 순위 함수

#### ■ Top 함수

##### ■ 질의 결과 튜플 수의 제한

- 질의 결과는 ORDER BY 절의 이용하여 정렬할 수 있음
- ORDER BY 정렬 기준에서 특정 등수 / 비율까지만 보고 싶은 경우
- Top(n) 함수를 이용함

```
SELECT TOP(n) 속성명  
...  
ORDER BY 속성명
```

- 동률이 있을 때 모두 보고 싶은 경우
- WITH TIES를 사용함

```
SELECT TOP(n) WITH TIES 속성명  
...  
ORDER BY 속성명
```

##### ■ 정렬 기준 특정 비율까지만 보고 싶은 경우

- Top(n) PERCENT를 이용함
- 상위 n%까지만을 출력하게 됨
- WITH TIES와도 같이 쓸 수 있음

```
SELECT TOP(n) PERCENT [WITH TIES] 속성명  
...  
ORDER BY 속성명
```

# 핵심요약

## 1. 순위 함수

### ■ RANK 함수

- TOP() 함수를 쓰면 결과 수를 제한함
- TOP() 함수는 등수를 구할 수 없음
- RANK 함수 : 각 튜플에 등수를 표시함

```
RANK함수 over (order by 속성명 [asc|desc] )
```

### ■ RANK() 함수

```
SELECT 속성명, RANK () OVER (ORDER BY 속성명 [asc|desc] )
```

- 동렬에 대하여 동일 등수 배정
- 비연속식 등수 배정

### ■ DENSE\_RANK() 함수

```
SELECT 속성명, DENSE_RANK () OVER (ORDER BY 속성명)
```

- 동렬에 대하여 동일 등수 배정
- 연속식 등수 배정

### ■ ROW\_NUMBER() 함수

```
SELECT 속성명, ROW_NUMBER () OVER (ORDER BY 속성명)
```

- 동렬에 대하여 임의 등수 배정
- 연속식 등수 배정

### ■ NTILE(n) 함수

- 전체 튜플을 num개로 균등 분할하여 순위 지정



# 핵심요약

## 2. 그룹 별 순위

### ■ 그룹 별 순위 지정

- 기존 RANK함수 문법
  - 전체 결과에 대한 속성값 기준 등수 지정이 됨
- PARTITION BY 속성명
  - 튜플들을 속성값에 따라서 그룹핑함
  - 각 그룹에 대하여 순위 함수를 적용함

```
RANK() over (PARTITION BY dno ORDER BY salary desc)
```

- 그룹별 특정 등수의 정보를 보고 싶은 경우
  - WHERE 절을 같이 활용함

```
RANK() over (PARTITION BY dno ORDER BY salary desc) AS 속성명  
...
```

```
WHERE 속성명 = 등수
```

### ■ 그룹 별 집단 함수

- 그룹 별 집단 함수의 적용

```
SELECT 집단 함수 ~ GROUP BY~
```

- PARTITION BY를 이용해서도 그룹 별 집단 함수를 적용할 수 있음

```
SELECT 집단함수() OVER (PARTITION BY 속성명)
```

### ■ 행 순서 함수

- 정렬된 대상에서 특정 순위의 튜플들을 추출할 필요가 있을 때 사용되는 함수
- FIRST\_VALUE 함수
  - 정렬 대상에서 첫 번째 데이터 추출
- LAG / LEAD 함수
  - 지정된 순서에서 선행 / 후행 데이터를 참조하는 함수