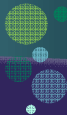


제3장

SQL 기초-2



5. 윈도우 함수의 개념

- 윈도우 함수(window function)
 - 테이블의 행과 행 사이 관계를 쉽게 정의하기 위해 MySQL에서 제공하는 함수
 - OVER 절이 들어간 함수
- 윈도우 함수와 함께 사용되는 집계 함수
 - AVG(), COUNT(), MAX(), MIN(), STDDEV(), SUM(), VARIANCE() 등
- 윈도우 함수와 함께 사용되는 비집계 함수
 - CUME_DIST(), DENSE_RANK(), FIRST_VALUE(), LAG(), LAST_VALUE(), LEAD(), NTH_VALUE(), NTILE(), PERCENT_RANK(), RANK(), ROW_NUMBER() 등

6. 순위 함수

■ 순위 함수

- 결과에 순번 또는 순위(등수)를 매기는 함수
- 비집계 함수 중에서 RANK(), NTILE(), DENSE_RANK(), ROW_NUMBER() 등이 해당
- 순위 함수의 장점은 구문이 단순함.
- 코드가 명확하고 수정이 쉬우며, 효율성도 뛰어나
- DB엔진의 부하를 최소화

```
<순위함수이름>() OVER(  
  [PARTITION BY <partition_by_list>]  
  ORDER BY <order_by_list>)
```

6. 순위 함수(실습 내용)

1 sqldb 초기화하기

1-1 열린 쿼리 창을 모두 닫고 새 쿼리 열기

2 키가 큰 순으로 정렬하기

2-1 회원 테이블(userTBL)에서 키가 큰 순으로 순위 매기기

```
USE sqldb;  
SELECT ROW_NUMBER() OVER(ORDER BY height DESC)  
"키큰순위", userName, addr, height  
FROM userTBL;
```

	키큰순위	userName	addr	height
▶	1	박수홍	서울	183
	2	강호동	경북	182
	3	이희재	경기	180
	4	남희석	충남	180
	5	유재석	서울	178
	6	김용만	서울	177
	7	신동엽	경기	176
	8	김제동	경남	173
	9	김국진	서울	171
	10	이경규	경남	170

6. 순위 함수(실습 내용)

2-2 키가 같은 경우 이름의 가나다순으로 정렬

```
SELECT ROW_NUMBER() OVER(ORDER BY height DESC,  
    userName ASC) "키큰순위", userName, addr, height  
FROM userTBL;
```

	키큰순위	userName	addr	height
▶	1	박수홍	서울	183
	2	강호동	경북	182
	3	남희석	충남	180
	4	이희재	경기	180
	5	유재석	서울	178
	6	김용만	서울	177
	7	신동엽	경기	176
	8	김재동	경남	173
	9	김국진	서울	171
	10	이경규	경남	170

2-3 각 지역별로 순위 매기기

```
SELECT addr, ROW_NUMBER() OVER(PARTITION BY addr  
    ORDER BY height DESC, userName ASC) "지역 별키큰순위",  
    userName, height  
FROM userTBL;
```

지역별로 분류를 하고자 하면 **PARTITION BY** 구문을 사용하자.

	addr	지역별키큰순위	userName	height
▶	경기	1	이희재	180
	경기	2	신동엽	176
	경남	1	김재동	173
	경남	2	이경규	170
	경북	1	강호동	182
	서울	1	박수홍	183
	서울	2	유재석	178
	서울	3	김용만	177
	서울	4	김국진	171
	충남	1	남희석	180

6. 순위 함수(실습 내용)

2-4 키가 같을 경우 동일한 통수로 처리

```
SELECT DENSE_RANK() OVER(ORDER BY height DESC) "키큰순  
위", userName, addr, height  
FROM userTBL;
```

DENSE_RANK()는 값이 같은 경우 동일한 통수로 처리함.

	키큰순위	userName	addr	height
▶	1	박수종	서울	183
	2	강호동	경북	182
	3	이휘재	경기	180
	3	남희석	충남	180
	4	유재석	서울	178
	5	김용만	서울	177
	6	신동엽	경기	176
	7	김재동	경남	173
	8	김국진	서울	171
	9	이경규	경남	170

2-5 3통 다음에 4통을 빼고 5통이 나오게 하려면 RANK() 함수 사용

```
SELECT RANK() OVER(ORDER BY height DESC) "키큰순위",  
userName, addr, height  
FROM userTBL;
```

	키큰순위	userName	addr	height
▶	1	박수종	서울	183
	2	강호동	경북	182
	3	이휘재	경기	180
	3	남희석	충남	180
	5	유재석	서울	178
	6	김용만	서울	177
	7	신동엽	경기	176
	8	김재동	경남	173
	9	김국진	서울	171
	10	이경규	경남	170

6. 순위 함수(실습 내용)

2-6 전체 인원을 키가 큰 순으로 정렬한 후 몇 개의 그룹으로 분할

```
SELECT NTILE(2) OVER(ORDER BY height DESC) "반번호",  
       userName, addr, height  
FROM userTBL;
```

NTILE(나눌 그룹의 개수)

	반 번호	userName	addr	height
▶	1	박수홍	서울	183
	1	강호동	경북	182
	1	이희재	경기	180
	1	남희석	충남	180
	1	유재석	서울	178
	2	김용만	서울	177
	2	신동엽	경기	176
	2	김재동	경남	173
	2	김국진	서울	171
	2	이강규	경남	170

2-7 3개 반으로 분리

```
SELECT NTILE(4) OVER(ORDER BY height DESC) "반번호",  
       userName, addr, height  
FROM userTBL;
```

	반 번호	userName	addr	height
▶	1	박수홍	서울	183
	1	강호동	경북	182
	1	이희재	경기	180
	2	남희석	충남	180
	2	유재석	서울	178
	2	김용만	서울	177
	3	신동엽	경기	176
	3	김재동	경남	173
	4	김국진	서울	171
	4	이강규	경남	170

7. 분석 함수(실습 내용)

1 특정 데이터와의 차이 값 구하기

1-1 회원 테이블(userTBL)에서 키가 큰 순으로 정렬한 후 다음 사람과의 키 차이 구하기

LEAD() 함수를 사용하면 다음 사람과의 키의 차이를 나타낼 수 있다.

```
USE sqldb;  
SELECT userName, addr, height AS "키",  
       height - (LEAD(height, 1, 0) OVER (ORDER BY height DESC)) AS "다음 사람과 키 차이"  
FROM userTBL;
```

	userName	addr	키	다음 사람과 키 차이
▶	박수훈	서울	183	1
	강호동	경북	182	2
	이회재	경기	180	0
	남희석	충남	180	2
	유재석	서울	178	1
	김용만	서울	177	1
	신동엽	경기	176	3
	김재중	경남	173	2
	김국진	서울	171	1
	이경규	경남	170	170

7. 분석 함수(실습 내용)

1-2 지역별로 가장 키가 큰 사람과의 차이 구하기

```
SELECT addr, userName, height AS "키",  
       height - (FIRST_VALUE(height) OVER (PARTITION BY addr ORDER BY height DESC))  
       AS "지역별 최대키와 차이"  
FROM userTBL;
```

	addr	userName	키	지역별 최대키와 차이
▶	경기	이희재	180	0
	경기	신동엽	176	-4
	경남	김제동	173	0
	경남	이경규	170	-3
	경북	강호동	182	0
	서울	박수홍	183	0
	서울	유재석	178	-5
	서울	김용만	177	-6
	서울	김국진	171	-12
	충남	남희석	180	0

7. 분석 함수(실습 내용)

2 누적 백분을 구하기

2-1 같은 지역의 회원과 비교하여 키가 크거나 같은 사람이 전체의 몇 %인지 누적 백분을 구하기

```
SELECT addr, userName, height AS "키",  
       (CUME_DIST() OVER (PARTITION BY addr ORDER BY height DESC))*100 AS "누적인원 백분율%"  
FROM userTBL;
```

	addr	userName	키	누적인원 백분율%
▶	경기	이희재	180	50
	경기	신동엽	176	100
	경남	김제동	173	50
	경남	이경규	170	100
	경북	강호동	182	100
	서울	박수홍	183	25
	서울	유재석	178	50
	서울	김용만	177	75
	서울	김국진	171	100
	충남	남희석	180	100

감사합니다.

