

# ch07. 다중행 함수와 데이터 그룹화

## 07-1. 하나의 열에 출력 결과를 담는 다중행 함수

그룹 함수 또는 복수행 함수로도 불리는 다중행 함수 *multiple-row function* 는 여러 행을 바탕으로 하나의 결과 값을 도출해내기 위해 사용하는 함수.

### 다중행 함수

함수	설명
SUM	지정한 데이터의 합 반환
COUNT	지정한 데이터의 개수 반환
MAX	지정한 데이터 중 최댓값 반환
MIN	지정한 데이터 중 최솟값 반환
AVG	지정한 데이터의 평균값 반환

### SUM 함수

SUM([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)  
[합계를 구할 열이나 연산자, 함수를 사용한 데이터(필수)])

- SUM함수를 분석하는 용도로 사용한다면 OVER절을 사용
- 

SUM([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)  
[합계를 구할 열이나 연산자, 함수를 사용한 데이터(필수)])  
OVER(분석을 위한 여러 문법을 지정)(선택)

```
SELECT SUM(COMM)
FROM EMP;
```

### 데이터 개수를 구해주는 COUNT 함수

COUNT ([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)  
[개수를 구할 열이나 연산자, 함수를 사용한 데이터(필수)])  
OVER(분석을 위한 여러 문법을 지정)(선택)

```
SELECT COUNT(*)  
FROM EMP  
WHERE DEPTNO = 30;
```

## 최댓값과 최솟값을 구하는 MAX, MIN 함수

MAX ([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)  
[최댓값을 구할 열이나 연산자, 함수를 사용한 데이터(필수)])  
OVER(분석을 위한 여러 문법을 지정)(선택)

```
SELECT MAX(SAL)  
FROM EMP  
WHERE DEPTNO = 10;
```

- 오라클 DB에는 날짜 및 문자 데이터 역시 MAX,MIN 사용가능

## 평균 값을 구하는 AVG 함수

AVG ([DISTINCT, ALL 중 하나를 선택하거나 아무 값도 지정하지 않음(선택)  
[평균을 구할 열이나 연산자, 함수를 사용한 데이터(필수)])  
OVER(분석을 위한 여러 문법을 지정)(선택)

```
SELECT AVG(SAL)  
FROM EMP  
WHERE DEPTNO = 30;
```

## 07-2. 결과 값을 원하는 열로 묶어 출력하는 GROUP BY절

## GROUP BY절의 기본 사용법

여러 데이터에서 의미 있는 하나의 결과를 특정 열 값별로 묶어서 출력할 때 데이터를 ‘그룹화’한다고 표현한다. SELECT문에서는 GROUP BY절을 작성하여 데이터를 그룹화할 수 있다.

```
SELECT  [조회할 열1 이름], [열2 이름], ..., [열N 이름]
FROM    [조회할 테이블 이름]
WHERE   [조회할 행을 선별하는 조건식]
GROUP BY [그룹화할 열을 지정(여러 개 지정 가능)]
ORDER BY [정렬하려는 열 지정]
```

```
SELECT AVG(SAL), DEPTNO
FROM EMP
GROUP BY DEPTNO;
```

```
SELECT DEPTNO, JOB, AVG(SAL)
FROM EMP
GROUP BY DEPTNO, JOB
ORDER BY DEPTNO, JOB;
```

## GROUP BY절을 사용할 때 유의점

- 다중행 함수를 사용하지 않은 일반 열은 GROUP BY절에 명시하지 않으면 SELECT절에서 사용할 수 없다는 것.

### 07-3. GROUP BY절에 조건을 줄 때 사용하는 HAVING절

- HAVING절은 SELECT문에 GROUP BY절이 존재할 때만 사용 가능
- 그룹화된 결과 값의 범위를 제한하는 데 사용한다.
- 각 부서의 직책별 평균 급여를 구하되 그 평균 급여가 2000 이상인 그룹만 출력

```
SELECT DEPTNO, JOB, AVG(SAL)
  FROM EMP
 GROUP BY DEPTNO, JOB
  HAVING AVG(SAL) >= 2000
 ORDER BY DEPTNO, JOB;
```

## HAVING절의 사용법

```
SELECT  [조회할 열1 이름], [열2 이름], ..., [열N 이름]
FROM    [조회할 테이블 이름]
WHERE   [조회할 행을 선별하는 조건식]
GROUP BY [그룹화할 열을 지정(여러 개 지정 가능)]
HAVING  [출력 그룹을 제한하는 조건식]
ORDER BY [정렬하려는 열 지정]
```



### HAVING절을 사용할 때 유의점

- HAVING절도 WHERE절 처럼 지정한 조건식이 참인 결과만 출력한다 는 점은 비슷하다.
- 하지만 WHERE절은 출력 대상 행을 제한하고, HAVING절은 그룹화된 대상을 출력에서 제한하므로 쓰임새가 다르다.

- WHERE절을 사용하지 않고 HAVING절만 사용한 경우

```
SELECT DEPTNO, JOB, AVG(SAL)
  FROM EMP
 GROUP BY DEPTNO, JOB
  HAVING AVG(SAL) >= 2000
 ORDER BY DEPTNO, JOB;
```

- WHERE절과 HAVING절을 모두 사용한 경우

```
SELECT DEPTNO, JOB, AVG(SAL)
  FROM EMP
 WHERE SAL <= 3000
 GROUP BY DEPTNO, JOB
  HAVING AVG(SAL) >= 2000
 ORDER BY DEPTNO, JOB;
```

